VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STAVEBNÍ
ÚSTAV STAVEBNÍ MECHANIKY

FACULTY OF CIVIL ENGINEERING
INSTITUTE OF STRUCTURAL MECHANICS

USING OPTIMIZATION'S ALGORITHMS BY DESIGNING OF STRUCTURES

VYUŽITÍ OPTIMALIZAČNÍCH ALGORITMŮ PŘI NAVRHOVÁNÍ KONSTRUKCÍ

DISERTAČNÍ PRÁCE
DOCTORAL THESIS

AUTOR PRÁCE Ing. Filip Fedorik
AUTHOR

VEDOUCÍ PRÁCE doc. Ing. Jiří Kala, Ph.D.
SUPERVISOR

BRNO 2013

Institute of Structural Mechanics
Faculty of Civil Engineering
Brno University of Technology
Czech Republic

ABSTRACT

The application of optimization algorithms in the design of many economical and industrial problems currently represents a significant assignment. The development of high-powered computers allows an application of difficult mathematical techniques and physical phenomena to simulate real problems with sufficient accuracy. The optimization techniques used in engineering designs are mostly represented by modified mathematical programming methods with extension of their usability.

The aim of the presented thesis "Using Optimization's Algorithms by Designing of Structures" is to analyze the applicability of optimization procedures which are available in the widely used computing system ANSYS in civil and mechanical engineering practice. The numerical analyses were performed within the frame of multi-extreme, one to three dimensional optimization problems, multi-dimensional problems expressed by minimizing the weight of a truss beam and efficient design of air gap location in wooden studs from the point of view of thermal features of the structure. The analyzed optimization processes are in plurality verified with accurate manual computing and graphical solutions and the accent is put on optimization methods' possibilities to improve robustness, efficiency and accuracy of the optimization algorithms in civil engineering problems' designs.

The optimization methods represent a suitable approach to improve the efficient design of a wide range of civil and mechanical engineering structures and elements. By combination of their advantages and FEM/FEA method it is possible to achieve very good results, although robustness of the solutions is not guaranteed. The robustness and accuracy of the procedure could be increased by competent exploration of design space and suitable selections of optimization methods' features.

KEY WORDS

Design optimization, design space, design variables, objective function, feasible and infeasible space, FEM/FEA

ABSTRAKT

Využití optimalizačních algoritmů představuje v dnešní době významnou součást při návrzích v mnoha ekonomických a průmyslových odvětvích. Vývoj výkonných výpočetních nástrojů umožňuje aplikaci náročných matematických postupů a fyzikálních jevů pro zpřesnění výpočtů a simulování reálného chování konstrukcí s dostatečnou přesností. Společně s nároky na efektivitu v rámci stavebních i jiných konstrukcí a prvků byly v rámci inženýrských programů, do jejich struktur, implementovány optimalizační algoritmy umožňující dosažení, nebo přiblížení se k optimu z hlediska daných podmínek. Tyto optimalizační techniky často představují metody matematického programování doprovázené určitými modifikacemi k rozšeření jejich využití.

Cílem předložené práce „Využití optimalizačních algoritmů při návrhování stavebních konstrukcí" je analyzování vhodnosti použití optimalizačních postupů, které jsou dostupné v široce využívaném výpočetním systému ANSYS v oblasti stavebního a strojního inženýrství. Numerické aplikace jsou provedeny v rámci multi-extrémní, jedno až tří dimenzionální optimalizační úlohy, vícerozměrné úlohy, jejíž cílem je minimalizování hmotnosti příhradové konstrukce a efektivní umístění vzduchových kapes v dřevěném sloupu s ohledem na tepelně technické vlastnost konstrukce. Analyzované optimalizační postupy jsou ve většině případů verifikovány s přesným výpočetním a grafickým řešením a důraz je kladen na zvýšení robustnosti, efektivnosti a přesnosti daných metod při navrhování běžných stavebních úloh.

Optimalizační postupy představují vhodný přístup ke zvýšení efektivnosti návrhů široké škály konstrukcí a prvků ve stavebním a strojním inženýrství. Kombinací optimalizičních algoritmů a metody MKP je možné docílit velmi dobrých výsledků, přestože není, u většiny případů, zaručena robustnost řešení. Robustnost a přesnost těchto postupů může být zvýšena náležitým průzkumem návrhové oblasti a vhodnou volbou charakteristik použité optimalizační metody.

KLÍČOVÁ SLOVA

Konstrukční optimalizace, návrhový prostor, návrhové proměnné, účelová funkce, přípustná a nepřípustná oblast, MKP

Bibliografická citace VŠKP podle ČSN ISO 690

FEDORIK, FILIP. *Using Optimization's Algorithms by Designing of Structures.* Brno, 2013. 179 s. Disertační práce. Vysoké učení technické v Brně, Fakulta stavební, Ústav stavební mechaniky. Vedoucí disertační práce doc. Ing. Jiří Kala, Ph.D.

Declaration

I hereby testify that I have worked on the presented thesis on my own, and all publications I have used are listed in the reference.

Brno 19.7.2013

Ing. Filip Fedorik

ACKNOWLEDGEMENT

The presented thesis has been carried out by the Brno University of Technology (BUT), Czech Republic, Faculty of Civil Engineering, Institute of Structural Mechanics in the period 2006-2013.

First of all I would like to express my special gratitude to my supervisor doc. Ing. Jiří Kala, Ph.D. for his long-standing supervision, forwarding experiences, knowledge and giving me the golden opportunity to accomplish this work.

I want to thank the School of Engineering, Oulu University of Applied Sciences for giving me the opportunity to progress with the thesis and use their implements to improve the quality of my work.

I am grateful for the opportunity to draw on the research and knowledge base from GACR P104/11/0703.

Also, I would like to thank my family and all my friends who have supported me a lot with the production and processing of this work.

CONTENT

1 INTRODUCTION AND BACKGROUND

The effort of each civil engineer is to design a structure or its part, which satisfies all constraints and conditions. It has to resist all outdoor and indoor boundary conditions so that the durability and the life-span of the structure are not reduced. The engineer deals with the difficult process of considering all requirements, conditions and limitations to reach a certain aim.

Often, the aim of a design is to minimize manufacturing and operating costs, weight of a structure, application of inaccessible materials, etc. A suitable way to reach a solution for such problem is to use optimization methods.

The application of optimization algorithms in designing in many economical and industrial fields currently presents a significant assignment. This is especially influenced by an effort to achieve an efficient design of a problem from the point of view of financial expenses. An efficient design of any problem is a very difficult process taking into account all factors which can or might influence own design, as well as existence of a real problem. Recently, a number of softwares have been developed which include optimization algorithms. They allow designers to make their work faster, and make it easier to achieve efficient designs of structures. At first the optimization techniques were used in planning of military strategies and in planning of economical processes but currently it can be seen that they are advantageous for use in design of industrial units. The greatest progress of these applications was started by application of optimization methods in the information technology branch, which also represents the start point for their later usage in mechanical and civil engineering.

Economical, ecological, architectural, safety and many other factors which are taken into account in designing require a solution which approximates reality and, it is almost impossible to solve most of these cases manually without using modern computers and technologies. The available systems already allow, with sufficient accuracy, simulating material properties (changes in dependence on loading and boundary conditions) and difficult behaviour of structures in many static and dynamic situations, for example simulating of fluid or air flow around a structure, effect of seismic features, time dependent analyses, etc. With this quite a true copy of reality is obtained. This leads in achieving more accurate values of model properties with which it is possible to reach more efficient designs. The more accurate values of physical phenomena do not signify achieving an efficient design from the point of view of considering all affordable possibilities and conditions which are defined in structural designs.

Developing of high-powered computers allows an application of difficult mathematical techniques and physical phenomena to simulate real problems more precisely. Simultaneously with demand for searching optimal solutions there have been implemented optimization algorithms within the frame of engineering programs which can be helpful in reaching or getting near the required optimum.

The presented work “Using Optimization's Algorithms by Designing of Structures” deals with possibilities of using optimization techniques in efficient designs of structures or elements in civil engineering problems. By following theoretical findings of available optimization methods their efficiency is observed within the frame of the finite element method/analysis (FEM/FEA) computational procedures. Results from the optimization procedure present an initial design of a structure which is defined by lower (higher) value of predetermined goal subjecting to required conditions of the problem. After an initial mathematical computing model is built a mathematical expression of the problem is formulated based on stated optimization variables and then is subjected to an optimization procedure in order to achieve minimal or maximal objective function value.

If the aim of a structural design is its optimization or achieving efficiency by using an optimization method, the designer is referred to mathematical optimization techniques which have been developed within the frame of the Operating Research (Analysis). A mathematical expression of a real system has to be created according to a form which is required by a given method. In difficult or detailed structural problems this step becomes very complicated or nearly unachievable, and the designer is referred to his/her intuitions or analyses of several model variations followed by searching for the most suitable one. This could be solved by application of optimization techniques into computing systems which are designed for simulations of difficult structural problems. A widely used and successfully verified method which is usually used for complicated analyses is the finite element method (FEM/FEA).

Optimization methods were primarily developed for solving different problems than efficient designs of structures. However, some of the techniques are applicable also in mechanical and civil engineering problems. Most of the methods are able to solve only specific problems’ formulation which they were developed for. Even so, each problem requires specific manipulation. The aim of optimization techniques’ implementation into computing systems solving a broad range of technical or other problems is their universality.

The submitted thesis deals with a problem of an efficient design using optimization methods which are applied within the frame of structural models in the finite element method based program Ansys. It is divided into 10 chapters. The introduction discusses the general concept of optimization and its historic development. Recent usage of optimization techniques in practical problems, including a brief compendium of available literature concerning this problem is presented in chapter 2. Chapter 3 discusses about aims of the presented thesis.

Theoretical procedures of optimization methods and their generalization for practical problems are studied by Operating Research/Analysis and Mathematical Programming. For this reason chapter 4 briefly describes operating research methods and their applications and possibilities for solving optimization problems within the frame of mathematical programming.

Basic techniques of design optimization and terminology of extreme problems are outlined in chapter 5. Because of certain modifications of optimization methods which have to be done from the point of view of their universal applications, chapter 6 describes their general forms. For successful application of optimization methods in efficient structural design the designer should globally understand theoretical basics of used methods. These are given in chapter 7. One of the manners of efficient use of optimization methods in difficult structural or mechanical problems which are specified by multi-variables is their implementation into a computing system which is designed for this purpose. An example of such software is the multi-physical program Ansys whose part is also an optimization module Design Optimization. The procedure in creating of extremal problems within the frame of FEM/FEA system Ansys is different than traditional mathematical and structural analyses. Mutual relations between optimization techniques and multi-physical system based on finite element method are briefly explored in chapter 8. Chapter 9 deals with a set of case problems and their numerical solutions using ANSYS/Design Optimization program in different problems in civil engineering. The thesis's outputs are discussed and summarized in chapter 10.

1.1 HISTORY OF OPTIMIZATION PROCESSES

The first applications of mathematical optimization were discovered together with commencement of Greek mathematicians who started to use optimization for solution of their geometrical problems. For example Euclid (300 bc) considered the minimum distance between points of a line and he proved that the square's largest area is between rectangles with certain total length of edges. Since then, the development of optimization went through specific profession fields, mostly based on the experiences of the concerned individuals.

The development of optimization methods is traced to the days of Newton (†1727), Lagrange (†1813) and Cauchy (†1857). Primarily along the works of Newton and Leibnitze (†1716) there was already a possibility to develop computing optimization methods. The basics for computing of minimization of a function were brought by Bernoulli (†1705), Euler (†1783), Lagrange and Weierstrass (†1897). The first method for constraint optimization problems holds the name of its author, the Lagrange method. Then Cauchy came up with the first usage of method of steepest descent to solve unconstrained optimization problems. In spite of these early contributions, the progress in development of optimization during half of the 20th century was quite slow, but it changed after the coming of high-performance computers able to solve iterative and optimization processes, which were basics for further evolution of optimization methods. Then, significant expansion of literature came, which contains many optimization algorithms. In 1947 Georg Dantzig (†2005) determined a method to solve linear programming problems called Simplex method, and 10 years later a mathematician, Richard Ernest Bellman (†1984), came out with the basics for solving Dynamic programming technique. Kuhn and Tucker in 1951 established comprehensive conditions for solving of optimization programming, which became as basics for development in nonlinear programming. Valuable benefits for optimization method evolution in nonlinear programming were also brought by Zoutendijk and Rosen in the early 60s in the 20th century. Works by Carroll, Fiacco and McCormick facilitated solving of many difficult nonlinear programming problems, along with use of well known unconstrained optimization techniques.

Geometric programming was developed by Duffin, Zener and Peterson in the 1960's. The innovator work in the frame of integrated programming was developed by Ralph E. Gomory. It is one of the most popular areas in optimization. The reason is wide usage in solving of well known optimization problems. Dantzig, Charnes and Cooper developed techniques for stochastic programming and they solved problems with independent design variables along normal distribution.

Currently, there is placed more pressure on solving multi-objectives problems instead of problems with one objective function. The aim of programming is a technique to solve certain type of multi-objective optimization problems. At the beginning was the programming designed for solution of linear programming by Charnes and Cooper in 1961. In 1928 von Neumann put basics to game theory and since then, they are used in mathematical economic problems and in the military.

Since the very beginning optimization techniques have been passing through considerable progress and their application has expanded into many economical, medical and engineering specializations with a wide range of applicability.

2 OPTIMIZATION APPLIED TO MODERN CIVIL ENGINEERING PROBLEMS

After the very beginning a number of methods which allow finding extremes of functions with advantages were determined. Optimization in its broadest sense can be used to search an extreme in any engineering problem. The general formulation of optimization designing is handled by Operating Research (Analysis) [10], [11], [35], [36], [42], [49], [55]. The Operating Research deals with a wide concept of heterogeneous optimizations' applications [41]. Most frequently in the civil engineering problems are the Operating Research methods used to minimizing costs, weights of structures, etc. The aims of an optimization problem are quite often closely associated with each other. For example, consider the weight minimization which is reached by geometry changes. It doesn't mean that costs of the design are the same or lower than in an initial design, but the new geometry might lead to startlingly high costs. This could be affected by expensive procedures which would need to be performed on a construction site. Weight minimization is frequently used in the aircraft industry. In civil engineering the weight minimization is discussed especially in designing of structures which are subjugated to earthquakes, strong wind and/or another imposed load.

The optimization processes are very often applied in batch production where even subtle effectiveness of one element can lead to considerable savings within the frame of a global production. This could be found especially in the machine industry in mechanical component design.

In the civil engineering the optimization techniques are usually applied in designing of frames, basements, bridges, towers, chimneys, dams, etc.

Of course, optimization procedures are used in many other professions besides the manufacturing processes [8], [13]. Examples include the in efficient manipulation with material on conveyers (elevators, conveyor belts, cranes, vehicles), selection of optimal manufacturing processes in tooling of metal, optimal design of electrical grids, designing and planning of manufacturing procedures' schedules, optimal design of control systems [4], planning optimal strategies [2] to maximize profit considering competition on the demand side, statistic data processing and creating of empirical models from experimental outcomes to achieve the most accurate expression of physical phenomena [50], [68], [69], etc.

These days there are two optimization fields which are used in civil and mechanical engineering, design optimization and topological optimization respectively. Topological optimization [30], [33] forms optimal distribution of material in surface or volume of determined problem considering appropriate boundary conditions [8]. Design optimization [45], [46] deals with application of mathematical programming methods in solving of practical problems.

Currently few authors are occupied with mathematical optimization problems and their consecutive applications in practical problems from a general point of view [11], [37], [40],

[55], [65], but more often publications are focused on specific technical problems [1], [13], [15], [26], [29], [32], [43], [59], [70]. A subdomain of Operating Research which deals with optimization methods is mathematical programming [28], [44]. There are many optimization techniques and approaches [16], [27], [66] to solve practical optimization problems which can be classified as: linear and nonlinear optimization and constrained [19], [23] and unconstrained optimization problems [62]. The linear optimization [3], [11], [17], [18], [20], [25], [52], [67] techniques are applicable in problems where a mathematical model of a problem is formulated only by linear expressions. However, their utilization is found in many theoretical and practical applications. The nonlinear optimization [5], [10], [11], [14], [52], [55] deals with problems where at least one equation or inequation is expressed by nonlinear function.

Application of optimization methods in practical problems intervenes in many subcategories of mathematical programming, for example, the application of stochastic programming, integer programming, geometric programming or dynamic programming can be seen in [9], [10], [11], [21], [34], [55].

Certain efficiency in designing of civil and mechanical engineering problems has been achieved by using computing programs which allow the engineer to simulate real systems with sufficient accuracy. With these it is possible to violate or minimize many factors of safety. There are many available computing programs which can be used for simulation of a broad range of technical problems. Certainly the most famous and utilized technique for technical problems' simulations is the finite element method/analysis (FEM/FEA) [31]. Programs based on the FEM are already able to compute complex highly nonlinear dynamic analyses such as ADINA, ANSYS [76], COMSOL [77], ABAQUS, DYNA3D, etc. They are being often used by automobile, military, bioengineering, construction industries and also aerospace [64]. With accession of powerful computers and the computing programs, application of optimization techniques in practical engineering problems registered significant progress [54], [56]. Softwares which allow simulating complicated complex engineering problems made accessible to mathematicians implement optimization techniques into their structures. Currently, there are many engineering programs which closely cooperate with optimization algorithms and with these they are able to solve a broad range of optimization problems.

Lately, research which deals with applying of optimization algorithms in practical problems is quite often focused on special mathematical programs such as Matlab, Maple, Mathcad, MathWork, etc. They are already able to solve difficult complex optimization problems with high accuracy [60]. On the other hand their usage might be quite complicated in practical problems because they require exact mathematical formulation of a problem which could be a difficult task for a civil or mechanical engineer [71]. This leads engineer computing program developers to implement optimization algorithms into their structures so that it would be possible to use their techniques within the frame of performed simulations. The aim is implementing such algorithms which could be used for a wide range of theoretical and practical problems. The universality is usually obtained by combination of a few methods. Optimization processes are represented by iterative procedures of searching minimum or maximum of a final function (objective function). For this reason it is suitable to select algorithms which don't require difficult processing and computing time. For this purpose unconstrained optimization methods are often used, although most engineering problems are subjected to limiting conditions. This is then solved by applying penalty functions [6], [10], [55], [63] which insert constraints into an unconstrained optimization problem [3], [12], [55]. In engineering problems it is usually very difficult to express a real system by mathematical formulation. For this reason dependent variables' functions are approximated by an approximation method, where one of the most used is the method of least squares [7], [47], [48]. Then the approximated formulation of the real system is solved by either the direct or indirect optimization method [12], [39], [72], [74].

Considerable progress is registered in using modern methods of mathematical programming. They are usually referred to as heuristic optimization methods or evolutionary optimization methods. Their aim is applying methods which simulate adaptive system in a natural evaluation or natural facts of swarming behaviour in insects or schooling in fish into optimization techniques. These methods generally do not guarantee finding a problems' optimum but their use usually achieves very good results. The most discussed evolutionary computation algorithms are particle swarm optimization, genetic algorithms, simulated annealing and methods based on neural networks which represent a new generation among mathematical programming procedures [22], [51], [53], [61], [73].

3 MOTIVATION AND AIMS OF THE THESIS

Optimization processes of the operating research were originally developed more for applications in different professional fields than for efficient design of civil or mechanical engineering problems. The discipline of mathematical programming offers a set of optimization algorithms which are able to effectively search extreme of specific optimization problems. Especially in current research within the frame of modern methods of mathematical programming such as genetic algorithms, simulated annealing and methods based on neural-network, expressive progression occurs. But these algorithms are approachable with difficulties for most of the actual designers. A broad range and complexity of the structural engineering design problems and achieving of their mathematical expressions lead developers to implement universal optimization methods into widely used computing systems. These methods are usually caused by modifications of original methods of linear and nonlinear programming.

One of the most-used computing methods to analyze and simulate a wide range of complex engineering problems is the Finite Element Method. The FEM/FEA method simulates models of real systems by dividing them to small mutually connected elements, where each of them is determined by physical and mathematical expressions of the actual problem. Combination of the FEM/FEA method and optimization techniques could be a suitable way to achieve efficient design. Optimization techniques used in engineering problems are usually iterative procedures which require optimization problems' definition expressed by optimization variables. The optimization variables are applied and recomputed in each loop of the procedure till a convergence of the solution is achieved. All the variables which represent features of the problem must be parameterized. The parameters create individual terms of the optimization problem definition. In the design of efficient structures, the numerical model, solution and optimization problem definition must be combined so that divergence or mistaken results are avoided. It is a complex problem placed on the shoulders of the designer.

The aim of the presented thesis is to analyze the applicability of optimization procedures which are available in widely used computing systems in civil engineering practice. The analyzed optimization processes are in their plurality verified with accurate manually computed solutions or by using specialized systems such as optiSLang, Matlab, Maple, Mathcad and Comsol, and the accent is put on optimization methods' possibilities to improve robustness, efficiency and accuracy of the optimization algorithms in civil engineering problems' designs.

Application of optimization techniques in improving efficiency of structural designs is an independent category of the design which is set in complex engineering problems. Within the frame of the presented work two optimization methods, based on different approaches of the extremes searching, are analyzed. These methods are implemented in Design Optimization module which is an individual component of the widely used multi-physical system ANSYS.

4 OPERATING RESEARCH

The field which deals with a methodology of optimization techniques of mathematical methods, is referred to as Operating Research (or Operating Analysis) [35], [36]. Its aim is to create optimal decision and modelling of deterministic and probabilistic systems, which proceed from real-life. The systems for suitable decisions are found in many fields, for example military, business, economic activities, engineering and natural and social science. Mostly, they are characterized by a requirement to establish reduction of invested costs. Operating research considers present system behaviours and their applications, where it evaluates their efficiency and designs a method for the next process. It integrates accesses to solve wide decision-making problems established on mathematical modelling. Among methods which are able to solve operating analysis problems are the mathematical programming techniques, stochastic process techniques and statistical methods. In designs within the engineering field, one of the methods from mathematical programming techniques is mostly used to create more efficient structures using optimization algorithms. A process which has to be done along application of operating analysis to solve an engineering problem is divided into the following mutually connected phases.

1. The first step is to define the real system which will be subjected to analysis. It is necessary to involve all conditions and circumstances which are loading the system, or which could influence its behaviour in any way. These have to be part of the design to cover reliability and life-span of the structure. For example, sporadic loads from a strong wind, singular forces, or even biological effects such as moulds.
2. The second step is building a model and controlling accuracy. If the aim of the operating analysis is to investigate a structural element or a structure from the point of view of tenseness, deformation, thermal or other analysis, it is advisable to verify the model before the optimization technique is used. This means that the problem is solved without an optimization technique and verification of accuracy of the solution.
3. Because the operating analysis is expressed by mathematical techniques to solve a real system, it is necessary to formulate a physical model via mathematical expression = mathematical model.
4. Next, the mathematical model is subjected to an optimization process using some of the techniques which lead to minimizing or maximizing of the objective function.
5. When the optimization process is done, outcomes have to be evaluated. In the operating analysis processes the outcome is usually a set of results obtained from each performed iteration. Then, only that outcome which satisfies all the defined conditions and at the same time shows the lowest or highest value of the objective function is chosen. If all the conditions and constraints are satisfied and the set reaches the minimal or maximal value of objective function, it is considered as optimum.

6. The last step is an implementation of obtained data to the real system.

If all the previous steps are satisfied, that is the achievement of the optimum and the implementation of obtained data to real system, it can lead to a more efficient design of the initial structure.

4.1 METHODS OF OPERATING RESEARCH

There are many problems, where optimization methods should be used. But, each problem usually has a different structure of the mathematical model, which means that it is impossible to use only one algorithm. For that reason, the problems of the Operating Research are classified into sections [35], [41]. Three main sections can be described. Their differences are in differing access to the problem in question. The classification is given as follows:

- Mathematical programming – is the widest discipline of operating analysis, because of the practicality of its method. It is also the most used in civil engineering. They can be applied to cases where the aims are to reach an efficiency of a structure.
- Stochastic process techniques – the goal of the theory is to optimize system large-scale service from the point of view of maximizing the profit and maximizing the satisfaction of customers.
- Statistical methods – it is a discipline which solves conflict decision problems. The goal is to analyze a problem where many subjects with conflicting meanings and the game theory search for the optimum strategy.

4.2 MATHEMATICAL MODEL

A fundamental step of optimization problems is to create a mathematical model, which is represented by a simulation of a real system. The mathematical model describes the real system by mathematical and physical expressions.

One of the most accurate methods to simulate a real system is to create an experimental model and subjugate it to laboratory tests. However, this process usually requires time-consuming tests with indispensable financial means which have to be expended within the experiments.

The next method to simulate a real system is to make a mathematical-physics model by one of the available programs which are designed for solution of difficult structural problems. The most widespread method for simulation of real systems is the Finite Element Method (FEM). There are many programs available based on the finite element method which are developed for broad range simulations of static and dynamic problems applicable in many different fields. Some of these programs have in their structures implemented optimization algorithms which can be used for an efficient design of analyzed problems. All the optimization tools which are used in efficient designs of structural problems have evolved from theoretical basics given by a science referred to as Operating Research or Operating Analysis.

According to the type of variables, which figure in the optimization process, the computational models could be categorized as follows:

- *Deterministic models* – the models include only deterministic (strictly given) values and expressions.
- *Stochastic models* – the models include at least one value which is a random variable, whereas a probability distribution of all variables in the model is known.
- *Strategic models* – the models include at least one quantity whose probability distribution is unknown, but only its lower and upper limits are defined.
- *Adaptive models* – some of the defined quantities in the model indicate incomplete information about their probability distributions. The information gets more accurate along simulation of a real system.
- *Fuzzy models* – the models include only quantities whose values are fuzzy sets or fuzzy members.

4.3 OPTIMIZATION PROBLEM

The aim of the optimization processes is to find an extreme point(s) of an objective function by suitable varying of optimization variables along all defined design conditions. A general optimization problem can be defined as follows:

$$\textit{Find } \boldsymbol{x}\textit{, which minimize } f(\boldsymbol{x}) \tag{4.3.1}$$

subject to:

$$\begin{aligned} g_i(\boldsymbol{x}) &\leq 0 \qquad (i = 1,2,3,\dots,m) \\ l_i(\boldsymbol{x}) &= 0 \qquad (i = 1,2,3,\dots,p) \end{aligned} \tag{4.3.2}$$

where $\boldsymbol{x}$ is a vector of design variables (section 5.3.9.1), g_i and l_i are limiting conditions of a design expressed by equalities or inequalities. The general optimization problem form is then changing as a consequence of objective function properties f, optimization variables x_i and design conditions g_i, l_i (section 4.4).

4.4 MATHEMATICAL PROGRAMMING

A designer whose aim is to use an optimization algorithm to reach more efficient structure design most often meets methods which are coming from the mathematical programming methods. Mathematical programming is one of the operating analyses disciplines which has been developed to find extreme(s) of multi-variables functions in multi-dimensional spaces and it searches points where the extremes are achieved [20]. Its goal is to include optimization procedures which allow finding an optimum of a function. With regard to a broad range of optimization problem definitions, mathematical programming is categorized as follows:

1. Linear programming
2. Nonlinear programming
3. Integer programming
4. Geometric programming
5. Dynamic programming
6. Stochastic programming
7. Modern methods of mathematical programming (genetic algorithms, neural networks, etc.)

4.4.1 Linear Programming

Linear programming is one of the mathematical programming methods which solves optimization problems where an objective function and limiting design conditions are expressed by linear functions only. The design conditions are defined by equalities, inequalities or a combination of the two. The general form of linear programming is as follows [23]:

$$Find\ \boldsymbol{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix},\ which\ minimize\ \ f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n \qquad (4.4.1)$$

subject to:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned} \qquad (4.4.2)$$

and

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ &\vdots \\ x_n &\geq 0 \end{aligned} \qquad (4.4.3)$$

where c_j, b_j and a_{ij}, while $(i = 1,2,\dots,m)$ and $(j = 1,2,\dots,n)$, are known constants and x_j are design variables (section 5.3.9). The general linear programming form assumes that it is possible to minimize the objective function $f(\boldsymbol{x})$, and the design conditions are defined by a system of equations or inequations. The next very important requirement is positive values of design variables. In the case that negative design variable values have occurred, certain modifications have to be done which lead to their positive expressions. Similarly, if the aim is to maximize the objective function, its general form (4.4.1) is expressed with an inverse sign. Then the problem leads to the following formulation:

$$\textit{Find } \boldsymbol{x}\textit{, which maximize} \quad -f(\boldsymbol{x}) = -c_1x_1 - c_2x_2 - \cdots - c_nx_n \tag{4.4.4}$$

The most widely known and used method of linear programming is undoubtedly *Simplex method* [44]. The simplex method simultaneously with increasing number of variables requires strong computational tools and a patient designer. Hence, the simplex method is subjected to further research and it goes through various modifications [57] for more efficient usage.

4.4.2 Nonlinear Programming

If at least one non-linear function (objective function or design conditions) is contained in the optimization problem, one of the nonlinear programming methods has to be used. Nonlinear programming is the widest discipline of mathematical programming. Almost each optimization problem could be considered as a certain case of nonlinear programming. In some special cases or if some modifications are performed, it is possible to solve engineering problem by a linear programming method, but the non-linearity is in most of them so considerable that the nonlinear programming method has to be applied. Hence optimization methods which are implemented in engineering programs designed for efficient designs of technical problems, are based on nonlinear programming algorithms. This makes them applicable universally to a broad range of engineering problems. A general formulation of the nonlinear programming problems is as follows:

$$\textit{Find } \boldsymbol{x}\textit{, which minimize } f(\boldsymbol{x}) \tag{4.4.5}$$

subject to:

$$\begin{aligned} g_i(\boldsymbol{x}) &\le 0 \qquad && (i = 1,2,3,\dots,m) \\ l_i(\boldsymbol{x}) &= 0 \qquad && (i = 1,2,3,\dots,p) \end{aligned} \tag{4.4.6}$$

where $g_i(\boldsymbol{x})$ are design conditions expressed by a system of inequations and $l_i(\boldsymbol{x})$ are design conditions determined by a system of equations. In the following table (Table 4.1) is a list of traditional *nonlinear programming* methods [55].

Table 4.1 Nonlinear programming methods

Elimination methods	Interpolation methods	Direct methods	Indirect methods
Fibonacci method *Golden section method* *Interval halving method*	*Quadratic interpolation method* *Cubic interpolation method*	*Random search method* *Grid search method* *Least square method*	*Steepest descent method* *Conjugate gradient method* *Newton's method*

4.4.3 Integer Programming

Most mathematical programming problems assume continuous functions in an interval of design variables. This means that a design variable value can reach any real number in the defined interval. However, there is a possibility to meet problems where the design variables values can be represented only by positive integer numbers. In this case, the problem is solved within the frame of the integer programming. The general integer programming problem can be stated in the following standard form [38]:

$$Find\,\boldsymbol{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}, \text{ which minimize } f(\boldsymbol{x}) \qquad (4.4.7)$$

subject to:

$$g_i(\boldsymbol{x}) \leq 0 \qquad (i = 1,2,3,\dots,m) \qquad (4.4.8)$$

where components (design variables individually) $x_1, x_2, \dots, x_n$ of vector $\boldsymbol{x}$ can be expressed only by positive integer numbers. In certain cases integer numbers may be required only for some design variables x_i values. Then the problem is referred to as *mixed integer programming*. *Pure integer programming* occurs when all design vector's components x are expressed only by positive integer numbers. Furthermore, specific optimization problems require only design variable numbers 0 and 1. Then the problem is solved by *zero-one programming*. Among methods solving the integer programming problems are, for example [38], [55]:

- Cutting plane methods
- Relaxations and bounds method
- Branch-and-bound method
- Dynamic programming methods
- Column generation
- Heuristic methods (local search methods, evolutionary algorithms, simulating annealing)

4.4.4 Geometric programming

Geometric programming (GP) forms a category of mathematical programming which solves optimization problems expressed by an objective function and design conditions in a certain shape. A process where a practical problem is transformed to the geometric programming formulation is generally referred to as GP modelling. If the transformation is performed and the GP formulation is obtained, the method is considered as very reliable [55]. In some practical optimization problems the transformation is very difficult or even impossible, thus GP cannot be used to solve it and find an optimum of an objective function. Then, it would be suitable to use one of the nonlinear programming methods [9], [55]. A general geometric programming problem consists of the following procedure:

$$s(\boldsymbol{x}) = c x_1^{a_1} x_2^{a_2} \dots x_n^{a_n} \qquad (4.4.9)$$

where $c > 0$ and $a_i \in \boldsymbol{R}$. Then, the function (4.4.9) is a monomial[1] of design variables $x_1, x_2, \dots, x_n$. A component c represents constant of a monomial and a_i is an exponent of a monomial. An exponentiated monomial by any exponent is still monomial. Thus:

$$s(\boldsymbol{x})^{\gamma} = (c x_1^{a_1} x_2^{a_2} \dots x_n^{a_n})^{\gamma} = c^{\gamma} x_1^{\gamma a_1} x_2^{\gamma a_2} \dots x_n^{\gamma a_n} \qquad (4.4.10)$$

A sum of more monomials in the following form:

$$r(\boldsymbol{x}) = \sum_{w=1}^{W} c_w x_1^{a_{1w}} x_2^{a_{2w}} \dots x_n^{a_{nw}} \qquad (4.4.11)$$

is referred to posynomial[2], where $c_w > 0$.
A general form of the geometric programming is as follows:

$$Find\, \boldsymbol{x},\ which\ minimize\ f(\boldsymbol{x}) \qquad (4.4.12)$$

subject to:

$$\begin{aligned} r_i(\boldsymbol{x}) &\le 1 \qquad && (i = 1,2,\dots,m) \\ s_i(\boldsymbol{x}) &= 1 \qquad && (i = 1,2,\dots,p) \end{aligned} \qquad (4.4.13)$$

where r_i are posynomials, s_i are monomials and $\boldsymbol{x}$ is a design variables vector.

1 Monomial - in the GP context is similar as standard definition of monomial in algebra, but in algebra, a monomial has the form (4.4.7), but the exponents a_i must be nonnegative integers, and the coefficient c is one. In the monomial in GP formulation the coefficient can be any positive number and the exponents can be any real numbers, including negative and fractional [63].

2 Posynomial (positive+polynomial) - posynomials are closed under addition, multiplication, and positive scaling. They can be divided by monomials [63].

4.4.5 Dynamic Programming

Dynamic programming solves problems of finding extreme points of functions which can be defined as a process control problem dependent on time. In the cases where time is not explicitly defined, the dynamic programming solution consists of additional implementation of time to a model. The keynote is to split up multi-dimensional optimization problem to a serial multistage decision processes. The individual stages of the dynamic programming procedure are connected from the beginning to the end thereby forming one complete unit, where the unit represents a real system of the optimization problem. The dynamic programming process usually requires more complicated preparation of the model. The reason is that besides a model formation, there is also a necessity to assess computing procedures, which have to simulate the real system.

The dynamic programming problem can be generally stated as follows [55]:

$$\textit{Find } \boldsymbol{x}\textit{, which minimize} \quad f(\boldsymbol{x}) = \sum_{i=1}^{n} U_i = \sum_{i=1}^{n} u_i(v_{i+1}, x_i) \qquad (4.4.14)$$

subject to:

$$v_i = t_i(v_{i+1}, x_i) \qquad (i = 1,2,\dots,n) \qquad (4.4.15)$$

where U is an objective function, v_i are incomes of a problem (state variables), t_i are transformation functions and $\boldsymbol{x}$ is a vector of design variables x_i. Dynamic programming is usually used in production planning for satisfying required conditions, extension of production, investment planning, etc [34].

4.4.6 Stochastic Programming

Stochastic programming deals with optimization problems, where we have optimization parameters instead of deterministic expressions, and they are formulated by stochastic, probabilistic or random variables [55]. The aim of stochastic programming is to transform a stochastic problem to a certain formulation of deterministic problem and then solve the problem using a linear, nonlinear, geometric or dynamic programming method. The main idea of stochastic programming is to find $\boldsymbol{x}$, which satisfies or satisfies with high probability the defined design conditions, and simultaneously an objective function value is small or small with high probability.

A general form of stochastic programming can be stated as follows:

$$\textit{Find } \boldsymbol{x}\textit{, which minimize} \quad f(\boldsymbol{x}) = \sum_{j=1}^{n} c_j x_j \qquad (4.4.16)$$

subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad (i = 1,2, \dots, m) \qquad (4.4.17)$$

$$x_j \geq 0 \qquad (j = 1,2, \dots, k) \qquad (4.4.18)$$

where c_j, a_{ij} and b_i are unknown random variables with known probability distribution.

In the case that some of the parameters change their values within the frame of an objective function, or competent design conditions, the problem is stated nonlinearly. Then, the general form of nonlinear stochastic programming is as follows:

$$Find\, \boldsymbol{x},\ which\ minimize\ f(\boldsymbol{y}) \qquad (4.4.19)$$

subject to:

$$P\left[g_j(\boldsymbol{y}) \geq 0\right] \geq p_j \qquad (j = 1,2, \dots, m) \qquad (4.4.20)$$

where $\boldsymbol{y}$ is a vector of n random variables $y_1, y_2, \dots, y_n$ including also design variables $x_1, x_2, \dots, x_n$.

4.4.7 Modern Methods of Mathematical Programming

During the recent decades a significant development has been made within the frame of optimization methods whose computational structure to find an extreme point of a problem differs from traditional mathematical programming methods (see above). They are often referred to as *heuristic optimization methods* or *evolutionary optimization methods*. These methods generally do not guarantee finding a problem optimum, but with their usage, very good results are usually achieved [22], [51], [53], [61].

4.4.7.1 Genetic Algorithms

One of the most significant heuristic methods is surely genetic algorithms (GAs), which are stochastic techniques based on computational analogy of adaptive system in a natural evolution. For this reason even the terminology that GAs use, is taken from biology. The basic foundations were introduced by Charles Darwin in 1859 and then Henry Holland in 1975. The GAs start with randomly chosen parent chromosomes which are coded vectors of control variables from the search space. The population is created by a set of parent chromosomes. By applying genetic operators which model genetic processes occurring in nature (selection, recombination and mutation) the population tends to improve chromosomes. Next, the chromosomes are compared, and the ones which will take part in the reproduction process are chosen. The selection is performed according to probability on the base of fitness function (normalized objective function). The fitness function helps to distinguish good and bad solutions. When the selection process is done a recombination and mutation is performed. Then the GP probabilistically or randomly selects suitable features from the population to be parents of the next generation. The mutation can be performed in several ways, for example: subtree mutation, size-fair subtree mutation, node replacement mutation, hoist mutation, shrink mutation, permutation mutation, etc [53]. By the mutation a new chromosome from one individual is created with defined probability. Then the offspring is inserted into the population and it replaces the parent chromosomes. The procedure is performed until optimization problem criteria are achieved. The genetic programming can be briefly summarized into the following steps:

- Creation of zero population. One population corresponds to a set of individuals which together form one generation.
- The most suitable individuals are chosen. The selection is performed by *crossing*, *mutation* and *reproduction*. Here, crossing consists of a combination of some features of individuals, mutation in random change of some of their features and reproduction carries individuals from one generation to another without any changes.
- The procedure continues until a defined maximum number of steps or a required feasible solution is achieved.

The scheme of the genetic programming procedure is illustrated by the following figure (Figure 4.1) [61]:

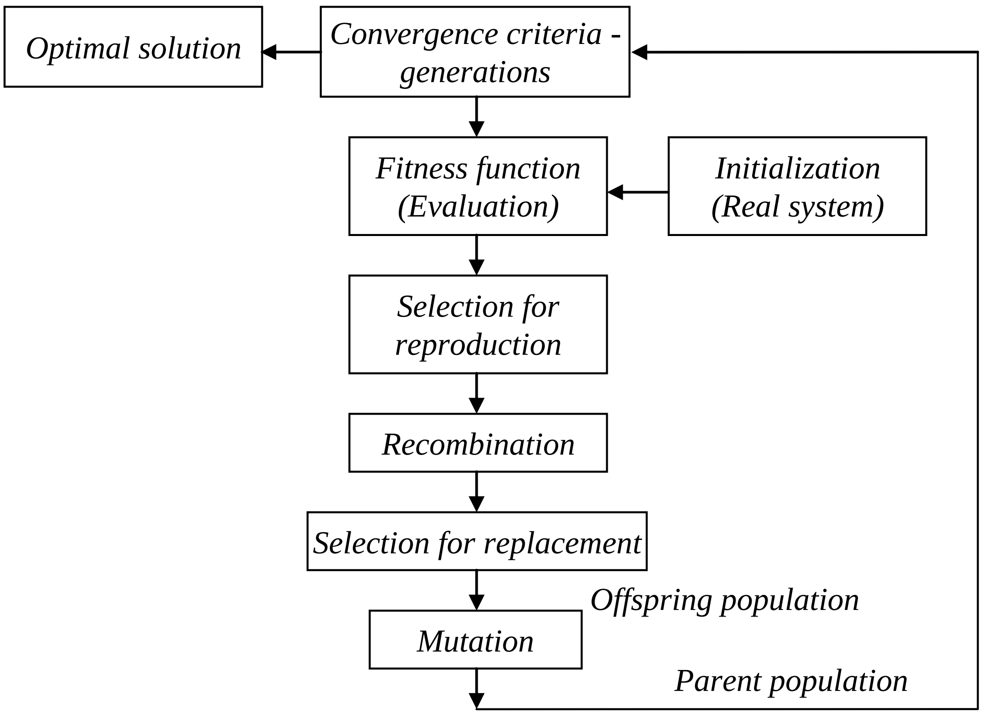

Figure 4.1 Scheme of Genetic programming procedure

The genetic algorithms in optimization problems do not guarantee an optimal solution. Inspite of this, experiences from many optimization problems using genetic algorithms achieve accurate and reliable results [24].

4.4.7.2 Simulated Annealing

Simulated Annealing is another heuristic method which allows to solve both discrete (non-continuous) and continuous problems of mathematical programming. It is based on the principle of thermal adaptation of steel (annealing). If the steel is melted by high temperature, atoms are able to freely move. Following the decrease of temperature their movement reduces, which brings about creation of crystals with minimal possible internal energy. For this reason the cooling must be followed out by a certain technique. In the case that the cooling is too fast, the crystals do not have enough time to form and the steel does not have the required properties. Therefore the cooling must be done slowly according to prescribed techniques. This process is referred to as *annealing*. The simulated annealing simulates the slow process of annealing while the minimum objective function value is achieved, where the temperature represents computational parameters, which are directed by concept of Boltzmann's probability distribution.

4.4.7.3 Particle Swarm Optimization

Particle swarm optimization simulates the behaviour of an insect swarm (ants, bees), bird swarm or fish swarm. Each member (insect, bird, fish) of the swarm is characterized by its location and speed in the n-dimensional space. They travel in a design space and remember the best location (objective function value). They exchange this information to each other and adapt it to their own location and speed.

4.4.7.4 Methods based on Neural-Network

Neural network is a computation procedure based on artificial intelligence which simulates behaviour of biological structures of live organisms. The network is assembled from neurons which form fundamental computational units. They exchange their information with each other. The amount of information depends on the complexity of neurons used, where each neuron has several incoming nodes but only one outcome. After processing the incoming data, the outcome information extends toward the end nodes.

5 DESIGN OPTIMIZATION

The following chapter presents distinguished items from the optimization processes, which are usually used in an efficient design of technical problems, where some of them are also presented via submitted work. The aim of the design optimization in civil or mechanical engineering is often based on minimization of elements' or structures' weight. A lower weight might lead to material quantity savings and then to decrease in capital investments. However, some cases with a lower weight can actually lead to increasing the investments. The higher investments can, for example be affected by unusual and difficult element geometry to build, or as the case may be more expensive a servicing during the life-span of the structure. The inferential point is that during the optimization design it is necessary to take into account all conditions, which can or might influence any part of that design. Some quantities are in reciprocal proportion and a designer has to decide which one is the more important to the prejudice of the other. A direct proportion of these factors can be reached by a new technology implementation and an application of the design optimization techniques in design of duplicate produced elements. If the design optimization techniques are to be used to design an efficient structure, the most important thing is to create a suitable mathematical model of the real system. Basic terms and necessary steps, which are an indispensable part of the design optimization techniques application, are the main topics of the following chapter.

5.1 OPTIMIZATION PROBLEMS CLASSIFICATION

Some civil engineering optimization problems make sense, if only integer values in the optimization process are used. An example could be the design of screw amount in a two elements connection, where it is necessary to achieve the integer values as the result of the optimization procedure, so the final objective function value has to be an integer. Most of these problems are solved by approximation of the discrete function (dependent functions - objective function, state variables) into a continuous one and then it is solved by one of the mathematical programming method for the continuous functions. When a convergence is achieved, the result values are rounded off to the nearer integer value (on the safety side). The obtained solution is then considered as the optimum of the problem. It is necessary to note, in some special cases this process might lead to a false solution [49]. Even so, this procedure can be used in efficient designs. Achieving of the optimum is not guaranteed, but there is an opportunity to get near to it. If it is still required to use a discrete optimization method to solve the integer problem, a general form of the optimization problem (eqs. 4.3.1 and 4.3.2) is extended within the following condition:

$$x_i \in Z, \qquad for\,all \qquad i$$

where Z denotes the set of all integers. These problems are included in the mathematical programming problems, through a subcategory called *integer programming* (section 4.4.3).

The problem is usually mentioned by discrete optimization where just one number is a result from a final design set, in contrast to the optimization algorithms which use continuous functions, where a set of vectors is usually searched with real numbers. The continuous function optimization is ordinarily considered simpler, because it is possible to obtain detailed information about a trend and function's form and trend of a smooth objective function in certain points of a design vector $\boldsymbol{x}$ (section 5.2.3) and their vicinity. It is not possible to achieve this information in case of the discrete problems, for reason of eventual diametrical differences between functional values of two reciprocal adjacent points. This information leads specialists to implement into civil and mechanical programs optimization algorithms which are established for solving the optimization, where dependent variables are expressed by continuous functions. These algorithms cannot guarantee the optimum achievement, but they are able to solve a broad range of technical as well as humanitarian problems. From this reason the submitted work is localized to efficient design with using the optimization algorithms instead of optimization of structures. In the following an explanation of the extreme searching terms is defined within the frame of continuous functions.

If the aim of the designer is to evaluate some physical quantity within the frame of a structural element, or an entire structure, the mathematical model has to be created. Afterwards, if the mathematical model should be subjected to the optimization procedure of purpose to achieve efficiency, it is necessary to define optimization variables. With the defined optimization variables the optimization process can be performed. The optimization variables are expressed by quantities which directly or indirectly come into the computation and their values influence the optimization process or its outcomes. Two main optimization variables are involved into the optimization processes. They are a *design vector* and an *objective function*. Then, the general formulation of optimization or mathematical programming can be expressed as:

$$\textit{Find } \boldsymbol{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}, \textit{ which minimize } f(\boldsymbol{x}) \qquad \textit{(5.1.1)}$$

subject to:

$$\begin{array}{ll} g_i(\boldsymbol{x}) \leq 0 & (i = 1,2,3,\ldots,m) \\ l_i(\boldsymbol{x}) = 0 & (i = 1,2,3,\ldots,p) \end{array} \qquad \textit{(5.1.2)}$$

where $\boldsymbol{x}$ is n-dimensional design vector and $f(\boldsymbol{x})$ denotes the objective function. $g_i(x)$ and $l_i(x)$ are design vector functions, which determine conditions of the design.

5.2 EXTREMES OF FUNCTION

Optimization is a process whose aim is to find an extreme(s) of a function, which presents an object of the defined problem that satisfies all conditions. The objective function has, in the neighbourhood of extreme points, a *convex* (for minimum) or *concave* (for maximum) shape. If the function is defined within the frame of an interval which could be expressed by design conditions, it is possible to achieve more than one extreme. Considering extreme properties, *local* or *global* extremes of the function can be reached. If a point is found which presents a minimal value of a function in a certain location, but does not reach the minimum within the frame of the entire interval, the local minimum is achieved. If it is to the contrary, it is the local maximum. A point is the global minimum or maximum, if its functional value is the minimum or maximum of the entire interval of the function. If the objective function is through its entire interval strictly convex or strictly concave, the local extremes are the global extremes at the same time. If the objective function trend is decreasing or increasing only very near to the boundary of an interval, the point is called *strict local minimum* or *strict local maximum*. A range for extreme searching of an objective function is determined by an interval, which is limited by predefined design conditions of a problem (in the case of constrained optimization problem described in section 6.3).

5.2.1 One-Dimensional Optimization Problem

The designer might meet a one-dimensional optimization problem in the civil and mechanical engineering field in a few isolated cases only. The one-dimensional optimization problem is defined by a minimization (maximization) of a function, which includes one variable only. There are many methods which can be used for analyzing solutions to this problem, and their process seems to be simple. Therefore it can quite often lead to very difficult expressions of mathematical functions which represent the optimization variables and it is more suitably dealt with using some accessible optimization tool. In practice it could be, for example, searching of an optimal distance between two supporting elements, where a mathematical expression might be very difficult, if the elements have untraditional shape and many load spectrums take effect the structure. Nevertheless, it is easier to understand the one-dimensional problem, from which most multi-dimensional optimization problem methods are developed. It allows to clearly observe a varying variable during the iterative optimization procedure. There is also sometimes a possibility to control the problem and the optimization process graphically. Hence, the following text contains the extreme searching problem terms and their description via the one-dimensional optimization problem. The main property of an extreme (critical point) is that the tangent orientation of a function is equal to zero in the extreme point (it is parallel with the x axis). Thus:

$$\frac{\partial f(x)}{\partial x} = 0. \tag{5.2.1}$$

The equation (5.2.1) expresses a sufficient condition for extreme existence of the function f. The roots x_i of the equation (5.2.1) present stationary points (the points where extremes may be founded). The stationary point may be a minimum, maximum, or inflection point. If additional information is needed the second derivative is performed. If the obtained function with x_i roots is positive, i.e.

$$\frac{\partial^2 f(x)}{\partial x^2} > 0 \tag{5.2.2}$$

the local minimum is achieved. Otherwise, if it is negative, i.e.

$$\frac{\partial^2 f(x)}{\partial x^2} < 0 \tag{5.2.3}$$

the local maximum is achieved. If the second derivative is equal to zero for x_i roots, i.e.

$$\frac{\partial^2 f(x)}{\partial x^2} = 0 \tag{5.2.4}$$

then it is necessary to use higher-order derivatives to make the final decision about the critical point character. However, it has to be noted that for example an exponential function acts distinctly with an even and uneven exponent. Hence, along a difficult function, it is suitable to explore a neighbourhood of the critical point to verify its correctness. If the following relation is reached, for the critical point on the functional interval I, then:

$$f(x) > f(x^*) \tag{5.2.5}$$

for all x within the frame of the interval I, a global minimum is achieved. If it is to the contrary a global maximum is achieved:

$$f(x) < f(x^*). \tag{5.2.6}$$

The basic concepts of extreme points finding are graphically represented in Figure 5.1, where a one-dimensional optimization problem is pictured through the following form:

$$Find\, x,\ which\ minimize\, f(x) = sin(0{,}6x) - cos(0{,}4x)$$

subjected to:

$$x \geq 2\pi$$

$$x \leq 9\pi$$

where the determined conditions define the interval I of the function f for the critical point searching simultaneously.

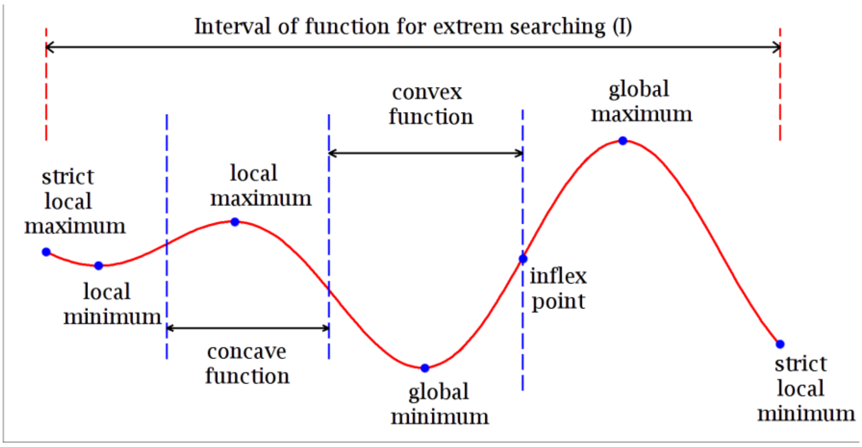

Figure 5.1 Basic concepts of extreme searching

5.2.2 Multi-dimensional Optimization Problem

According to the previous one-dimensional optimization problem definition, it is possible to come out with a determination for multi-dimensional optimization problems [3], [27]. To find critical points of two and more dimensional problems the direct and gradient method can be used. The direct methods (section 6.2.1) use approximations of functions and their functional values along the function to determine the extreme points. To find extreme points with using the gradient methods, derivatives of the function are used. Against the one-dimensional problem, where a gradient of the function f is a scalar quantity, in the multi-dimensional problems it is a vector. The main difference is in solving of partial derivatives instead of one-variable function derivatives.

The following criteria for extreme point presence assume $x_1, x_2, \ldots, x_n$ variables. Then, the direction vector $\nabla f(\boldsymbol{x})$ is defined by n-dimensional vector:

$$\nabla f(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{pmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \frac{\partial f(\boldsymbol{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\boldsymbol{x})}{\partial x_n} \end{pmatrix}. \qquad (5.2.7)$$

Accordingly as a one-dimensional optimization problem, the stationary points correspond to the zero first derivative of a function $f(\boldsymbol{x})$. Thus:

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} = 0 \qquad (i = 1,2, \ldots, n). \qquad (5.2.8)$$

The next decision, if the stationary points are also extremes or saddle points, the second partial derivatives of a function $f(\boldsymbol{x})$ are performed for an each variable x_i. A summation of all obtained partial derivatives is expressed by a matrix, well known as the Hessian matrix $\boldsymbol{H}$.

$$\boldsymbol{H}(\boldsymbol{x}) = \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}^2} = \begin{bmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 x_n} \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2^2} & & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n^2} \end{bmatrix} \tag{5.2.9}$$

Through the following conditions the point properties are obtained:

1. The local minimum in the point x_i is obtained, if:

$$\boldsymbol{H}(x_i) \geq 0 \tag{5.2.10}$$

   that is, the Hessian matrix is positive-semidefinite (the function $f(\boldsymbol{x})$ is convex near to the point x_i).

2. Strict local minimum is achieved in the point x_i, if:

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} = 0 \quad \textit{and simultaneously} \quad \boldsymbol{H}(x_i) > 0. \tag{5.2.11}$$

   In this case, the Hessian matrix is positive-definite and the function $f(\boldsymbol{x})$ is strictly convex near the point x_i.

3. Strict local maximum is achieved in the point x_i, if:

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} = 0 \quad \textit{and simultaneously} \quad \boldsymbol{H}(x_i) < 0. \tag{5.2.12}$$

   In this case, the function $f(\boldsymbol{x})$ is strictly concave near the point x_i.

4. The point x_i is a local saddle point of the function $f(\boldsymbol{x})$ if in this stationary point is not extreme and simultaneously the Hessian matrix $\boldsymbol{H}(x_i)$ is indefinite. Thus:

$$\boldsymbol{H}(x_i) < 0. \tag{5.2.13}$$

5.3 BASIC CONCEPTS OF DESIGN OPTIMIZATION

The current development in computing technologies allows designers to simulate a broad range of difficult engineering problems through computing programs. They are able to simulate and analyze difficult physical and chemical processes within the frame of a structure and its vicinity. A frequent aim of program developers is to implement optimization algorithms into the program structure so that they would be conductive in efficient designs of economical and/or technical problems. Researchers and computing developers make efforts to adapt some optimization techniques into a program that could be further universally used for a broad range of problems, which are being simulated by the certain program. If the designer's aim is to use an optimization algorithm in an efficient design of an element or structure with using an optimization method, it is necessary to create a mathematical model of the problem. The mathematical model, as a fundamental step, is a mathematical expression (by equations and inequations) of all parameters (section 8.4.2.1) which directly or indirectly influence the optimization procedure. The mathematical model is created according to properties of the real system. It usually depends on a geometrical model, which simulates the properties and geometry of the real system. The design optimization is based on mathematical optimization methods, which are applied in problems of a practical nature. This work deals with an application of optimization methods for an efficient design in civil engineering problems within the frame of a computing system, which is based on the finite element method (FEM). The process of the efficient design of an element or entire structure can be split up into two domains, design and optimization respectively. Then, the general procedure (Figure 5.2) of the efficient design of a civil engineering problem using an optimization algorithm in the frame of a computing program can be defined by the following steps [11], [12], [16], [75]:

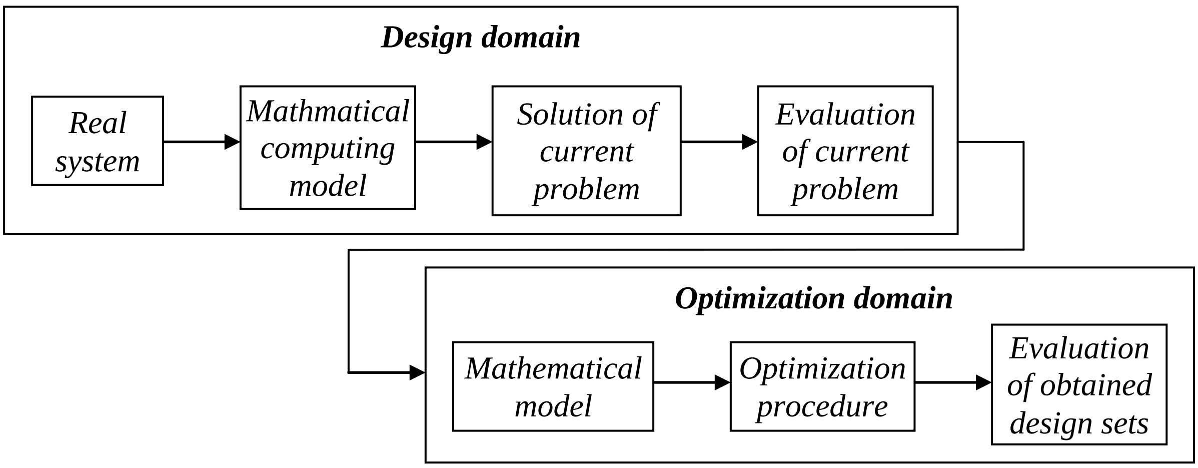

Figure 5.2 Design optimization procedure using computing system

5.3.1 Real System

The real system is an idea of a current or projected structure and its interaction with an environment. It is described in three-dimensional space and, almost always, by heterogeneous material properties. With this are also contextual internal and external loading conditions including natural phenomena, which influence the structure. If a simulation of the real structure is wanted, it has to be acceptable to use certain simplifications in physical and chemical processes, because the reality is very complicated and there is very small possibility to describe all design features precisely by mathematical expressions. In some cases, the design features can be measured on the construction site, or in a test-room, for example measuring deformation by stress tests, or measuring building physic quantities. But these procedures are usually very expensive and some of them even need very long time to be performed (for example: heat and moisture transfer simulation in building over a long period).

5.3.2 Mathematical Computing Model

The mathematical computing model is an expression of the real system by a computing program through three, two or one-dimensional geometrical diagrams, definition of physical phenomena surrounding the problem and defining material properties. Although the effort is to simulate the accurate real system, some simplifications are often used. The real system is always three-dimensional, but some cases allow to create only two or one-dimensional model to simulate the real quantity. For example, if the aim of the solution is to simulate heat transfer in a vertical wall, which does not change its material properties along the length and height of the wall, a one-dimensional model is adequate. If the homogeneity is defined in one direction only, the two-dimensional model is needed. If the geometry or material properties are varying in all directions, then a three-dimensional model has to be created to simulate the real system. For example, the usual simplification occurs in the definition of material properties, often without the knowledge of the designer, given that they consider constant values. But for most materials, the properties are varying with the influence of the external conditions. Certainly, these changes do not have to be so important in the solution, but they should be considered as the simplification.

5.3.3 Solution of Current Problem

Solution or analysis of the current problem consists in an evaluation of a design object, for example, the structural strength, building-physical features, dynamic behaviour, etc. The current problem analysis presents the initial point for the subsequent optimization procedure. For this reason, the solution of the current problem is very important, because it is the foundation for the entire optimization process (section 8.4). The optimization algorithms, used in the efficient design of complex technical problems, are quite often implemented into computing programs, which are based on the Finite Element Method. Then, the solution of the current problem consists of the entire finite element analysis.

5.3.4 Evaluation of Current Problem

The evaluation of the current problem is a very important step before the optimization procedure starts. Defined parameters and quantities from the solution of current problem obtained present the optimization parameters (section 8.4.2.1) in the optimization procedure. Part and parcel of the current problem evaluation should be verification with an independent method. A suitable way for the verification is to test the problem in a laboratory, or within the frame of the real system. If neither of them is available, the problem should be simplified so that it would be possible to calculate it manually. If both methods reach comparable outcomes, then the detailed outcomes are evaluated. The detailed evaluation is usually reached by numerical computing of the problem. There is a broad range of these quantities, which are usually searched before their parameterization and the following application in the optimization process. They can be, for example, deflection, strain, stress in stability analyses, temperature and heat flux in the thermal analyses, or other specific quantities corresponding to the performed analysis.

5.3.5 Mathematical Model

If the FEM (Finite Element Method) program is used, the previous steps present numerical mathematical model creation and one complete finite element analysis. The optimization is a mathematical procedure with the aim to achieve an extreme of a function. Thus, the numerical model (section 5.3.4) has to be expressed by parametric values which define the current problem. The optimization algorithms in FEA programs used are usually iterative mathematical processes, which change their parametric values while the optimization process is proceeding. After the initial design evaluation is done and the result parameters (volume of a structure, stress, temperature, etc.) are obtained, an optimization mathematical model is created. The optimization mathematical model is assembled from a set of simultaneous equations, inequalities or other mathematical functions, which all together present the objective function and design conditions (section 6.3.1) of the problem. The mathematical model forms fundamentals for solving operating research problems (chapter 4) and simultaneously expresses the real system of the problem.

5.3.6 Optimization Procedure

Then the mathematical model is subjugated to the optimization procedure. Corresponding to the form of functions, which represent the mathematical model (section 5.3.5), a suitable mathematical programming method (section 4.4) is chosen to minimize or maximize the objective function. The methods, which are objects of this work, are described in the chapter 7. With regard to a broad range of possible optimization algorithms, which are usually used for specific optimization problem, these have been developed for a multi-purpose utilization. This induces certain inaccuracies and there is no possibility to assure a robust design. For this reason, it is recommended to repeat the optimization process with different values of a current design, which represents an initial optimization point, or detailed exploring of a design space (section 7.3.10), for example via an optimization tool (section 7.3). The optimization procedure iterates till convergence criteria (sections 7.1.3 and 7.2.3) are achieved.

5.3.7 Evaluation of Optimization Procedure

The outcome of an optimization process is a group of design sets (section 5.3.11), where the number corresponds to a number of performed iterations plus the initial design. Each design set contains a list of optimization variables which have been a part of the optimization procedure. The design set which satisfies all defined design conditions and reaches the lowest (minimization), or the highest (maximization) objective function value from all of the others, is considered as an optimum. Whether the real optimum, from the mathematical point of view, is achieved depends on the robustness of the design and the detailed exploration of the design space (section 5.3.10). If the best obtained design set is considered as the best possible solution by a designer, the entire process ends. Then, the initial design parameters in the starting design are replaced by the final obtained parameters by which the efficient design of the current problem is obtained. If the best design set does not satisfy all the design conditions, or does not correspond to the optimum (reached by testing), the process is repeated with different initial design parameter values within the frame of the mathematical model, or there might be required different critical values of the optimization variables. Hence, it is recommended to verify the mathematical model, if the feasible extreme (section 5.3.12) exists according to the defined design conditions.

5.3.8 Lab Samples

The next option to simulate a reality is to make a lab sample and subjugate it to real measuring in a laboratory. Lab samples are usually small copies of a real system which has not yet been done. With these there is a possibility to analyze the real system with sufficiently accurate outcomes of measured data. For example, lab tests are often performed in a wind tunnel to simulate wind features in the vicinity of a measured element, lab tests of seismic conditions of a structure or other static or dynamic problems. In considering of an optimization, it is possible to avail oneself of lab tests, or measuring of a real system to create a mathematical model, which is then subjected to an optimization procedure. Also, lab tests could be used to verify optimization outcomes by simulating models which correspond to the final design sets.

5.3.9 Design vector

Each engineering problem, which is subjected to an optimization process, is defined by a set of variables (usually closely associated with each other) and their limiting values, which lead the entire design process. Some of them could be fixed values (such as dimensions according to a certain situation, accessible materials and their properties, etc.), which directly influence a problem evaluation, but they do not influence the optimization procedure itself. Searching for an extreme point of an objective function depends on suitable changes of design variable values. Generally, the design variables are presented in a design vector $\boldsymbol{x}$.

$$\boldsymbol{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \tag{5.3.1}$$

The design vector $\boldsymbol{x}$ is formed by components $(x_1, x_2, \dots, x_n)$, whose certain sequence and relations among them form an *objective function* [75].

5.3.9.1 Design Variables (DVs)

Design variables present independent variables in an optimization process. In engineering problems, they usually represent geometric dimensions of an element or a structure. Their values are changing in each iteration (optimization loop) during the entire optimization procedure, till the required computing conditions (minimum or maximum of the objective function, or other defined criteria) are achieved. The design vector is formulated as follows:

$$\boldsymbol{x} = \{x_1 x_2 x_3 \dots \dots \dots x_n\}^T. \tag{5.3.2}$$

All the design variables (DVs) must be limited from both sides. It means that the lower and upper limits are defined for each design variable:

$$\underline{x}_i \leq x_i \leq \overline{x}_i \qquad (i = 1,2,3, \dots \dots \dots, n) \tag{5.3.3}$$

where n is a number of design variables. If one or more limits are violated the infeasible solution is obtained (section 5.3.12).

5.3.9.2 State Variables (SVs)

Design of engineering problems usually requires certain conditions which have to be satisfied, such as reliability or life-span of a structure. The conditions are usually specified by standards (such as limit state designs), or by special demands on stability or stiffness of a structure. The state variables represent quantity variables which limit the problem of a design. They are functions of design variables, which means that the state variables change their values during an optimization procedure, along with changes of the design vector $\boldsymbol{x}$. State variables' limits are specified by lower and upper values, or they could be defined as one-side, which means they are limited only from below or only from above. Then a problem of the objective function minimization is defined as:

$$f = f(x) \tag{5.3.4}$$

subjected to:

$$g_i(x) \leq \overline{g}_i \qquad (i = 1,2, \dots \dots \dots, m_1) \tag{5.3.5}$$

$$\underline{h}_i \leq h_i(x) \qquad (i = 1,2, \dots \dots \dots, m_2) \tag{5.3.6}$$

$$\underline{w}_i \leq w_i(x) \leq \overline{w}_i \qquad (i = 1,2, \dots \dots \dots, m_3) \tag{5.3.7}$$

where g_i, h_i, w_i are state variables (SVs). Same as in the case of design variables, if one or more defined limits are violated the solution is infeasible (section 5.3.12).

5.3.10 Design Space

Most of the practical problems are subjugated to conditions which are specified by a designer or problems' regulations. In optimization, problems are usually defined by critical values of design variables (section 5.3.9.1) and state variables (section 5.3.9.2). All the critical values are usually known as optimization conditions or constraints (section 6.3.1). The optimization constraints separate feasible and infeasible spaces (section 5.3.12). The entire space where a computing is in progress is denoted as design space.

5.3.11 Objective Function (Obj)

The aim of each civil or mechanical designer is to find a solution which satisfies all defined conditions for a structure. Generally, there is more than one feasible solution and the goal of an optimization process is to find the best of them. A function which leads the optimization procedure and the aim of the process, which is to minimize its value in the operating analyses is *objective function* (*Obj*). Selection of the objective function depends on the particular problem analyses. Very often, it is defined by weight of the problem (especially in the aerospace industry), or by cost (in the engineering and economic sectors). Both of these cases lead to a minimization of the objective function. Certainly, the optimization processes do not solve minimization problems only, but also many engineering problems can lead to maximization of the objective function, for instance if the objective function represents production efficiency or stiffness of a structure. The objective function is in most cases evident. But the designer can meet problems where the optimization process with consideration of some criteria leads to final results which violate other criteria in the design. For example, in an efficient design of civil engineering structure the minimum weight of the structure need not correspond to the minimum tension and a minimum design tension need not correspond to a maximum natural frequency. The objective function choice becomes one of the most important steps in the optimization design processes.

Furthermore, some designs might require more than one criterion to satisfy, for example if there are requirements to minimize the weight and simultaneously to maximize stiffness of the structure. These designs where more than one objective functions are needed are known as *multi-objective optimization problems*. The multi-objective optimization problems can lead to certain difficulties where the objective functions might be in mutual conflict. Then the solution consists in defining one total objective function by a linear combination of all the objective functions which are specified in the problem. Then, if the particular objective functions are $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ the total objective function is as follows:

$$f(\boldsymbol{x}) = a_1 f_1(\boldsymbol{x}) + a_2 f_2(\boldsymbol{x}) \tag{5.3.8}$$

where a_1 and a_2 are constants, whose values represent relative importance of the one objective function against the other [55].

5.3.12 Design Set

The design set is a list of input and output parameters which directly figure in the optimization process. It involves design vector $\boldsymbol{x}$ and objective function values. An initial design is expressed by a set of originally defined optimization variables (design variables) which are known before the first optimization loop is performed. The remaining variables (state variables and objective function) are known after the initial iteration and evaluation is performed. The parameter values in the following iterations obtained depend on used optimization method or tool. A number of design sets depends on the optimization method, convergence criteria and a number of performed iterations. If all the conditions are or aren’t reached, the design sets can be classified as *feasible* or *infeasible* accordingly.

5.3.13 Feasible and Infeasible Solution

In the design optimization problems the designer usually meets inequalities which determine limit values of variables:

$$g_i(\boldsymbol{x}) \leq 0 \qquad (i = 1,2,\dots,n). \qquad (5.3.9)$$

Where $g_i(\boldsymbol{x})$ represents generally design variables and n a number of DVs. A set of values $\boldsymbol{x}$, which corresponds to equation:

$$g_i(\boldsymbol{x}) = 0 \qquad (i = 1,2,\dots,n) \qquad (5.3.10)$$

forms a certain $(n-1)$-dimensional design space, which is known as *constrained design space*. Thus the general design space is assembled from two regions:

$$g_i(\boldsymbol{x}) \leq 0 \qquad (i = 1,2,\dots\dots\dots,n) \qquad (5.3.11)$$

$$g_i(\boldsymbol{x}) > 0 \qquad (i = 1,2,\dots\dots\dots,n). \qquad (5.3.12)$$

Where all design sets, which satisfy the first condition (5.3.11), are *feasible design sets*. The final design sets corresponding to the condition (5.3.12) are called *infeasible design sets*. This means that a design set which violates at least one of the defined optimization problem conditions is an infeasible design set.

Limit values of optimization variables are usually considered with certain tolerances, which are internally set using a computing system, or they can be modified by a designer before a problem solution starts. Then the original minimization problem (5.3.4 - 5.3.7) considering the variables tolerances is formed:

$$f = f(x^*) \qquad (5.3.13)$$

where $\boldsymbol{x}^*$ is a design set defined as:

$$\boldsymbol{x}^* = (x_1^* x_2^* x_3^* \dots\dots\dots x_n^*) \qquad (5.3.14)$$

Then, the feasible sets are only obtained if the following conditions are satisfied:

$$g_i^* = g_i(x^*) \leq \overline{g}_i + \alpha_i \qquad (i = 1,2,3,4,\dots\dots\dots,m_1) \qquad (5.3.15)$$

$$\underline{h}_i - \beta_i \leq h_i^*(x^*) \qquad (i = 1,2,3,4,\dots\dots\dots,m_2) \qquad (5.3.16)$$

$$\underline{w}_i - \gamma_i \leq w_i^* = w_i(x^*) \leq \overline{w}_i + \gamma_i \qquad (i = 1,2,3,4,\dots\dots\dots,m_3) \qquad (5.3.17)$$

where α_i, β_i, γ_i are limit values tolerances of state variables, and

$$\underline{x}_i \leq x_i^* \leq \overline{x}_i \qquad (i = 1,2,3,4,\dots\dots\dots,n). \qquad (5.3.18)$$

In the case that a designer decides about unimportance of a violation of a limit value, or the size of the violation does not have any importance in the point of view of the entire design, even an infeasible design set might be considered as optimum of the optimization problem.

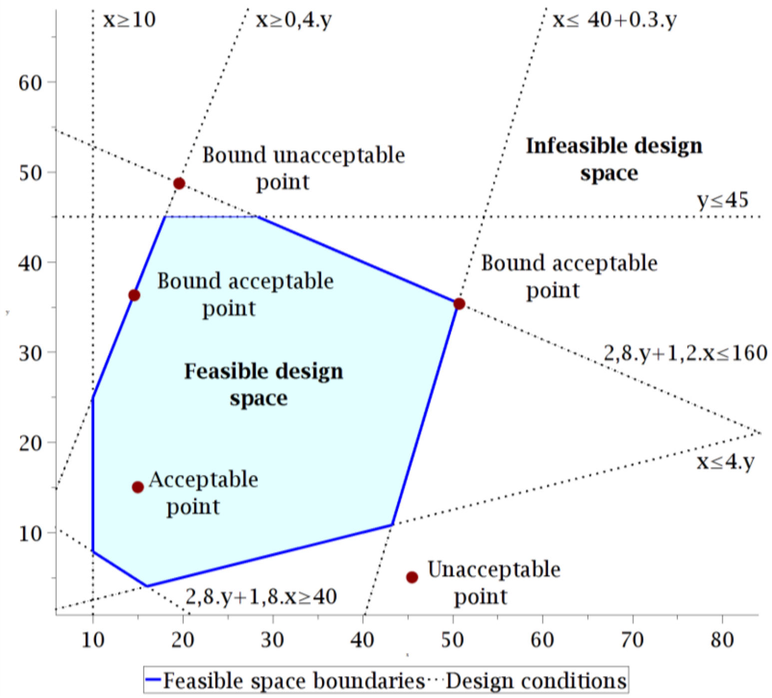

Figure 5.3 Feasible and infeasible design space

6 ORIGINAL METHODS OF DESIGN OPTIMIZATION

An optimization problem could be classified from many points of view, corresponding to real system characteristics, mathematical expression of the real system and a type of the solution. In the following material optimization problems are categorized in accordance with a number of optimization variables (one or multi-variable optimization), optimization strategy (passive or sequential optimization), method to search an extreme point of a function (direct or indirect methods), method to generate new points (deterministic or stochastic), and so on.

6.1 OPTIMIZATION PROBLEM CLASSIFICATION

Very importantly, classification of optimization problems is based on a number of phases, which are proceeded during computational procedure. The stages are expressed by two types of variables. They are design variables and state variables (section 5.3.9). The design variables define a system and lead its process during the entire optimization procedure. The state variables describe the behaviour of a whole system in any phase of the evaluation.

The design and state variables change their values in each optimization loop (iteration) till convergence is achieved by the prescribed manner. Another essential classification is based on the existence of design and state variable limits (constraints). Particularly in civil engineering problems, there is a sizeable chance of presence of limits, for example due to building standards or the demands of an investor. If there is a requirement for an objective function to satisfy at least one limit of any variable, it is considered as a *constrained optimization* problem [19], [23]. Otherwise, if the aim of the optimization procedure is to minimize or maximize the objective function without influence of any limit then the problem is classified as *unconstrained optimization* problem [23], [62]. Unconstrained optimization problems are generally considered as simpler because the constrained optimization problem solves not only a minimization or maximization problem, but also takes into account the constraints of a solved system. Hence, constrained optimization problems are often transformed to unconstrained optimization problems by use of certain processes.

6.2 UNCONSTRAINED OPTIMIZATION

Civil engineering problems which are subjected to a design require satisfying of all imposed conditions. There is a multitude of conditions occurring in the design. Each of them presents a certain limit on the design. The limits are not only financial, but also often of a design character. The most frequent are minimum and/or maximum dimensions of a structure element, limits of internal strain, deformation, etc. If an optimization algorithm is required to design such problems, an unconstrained optimization method has to be used.

Constrained optimization methods are generally considered less efficient, because in addition to minimization or maximization of the objective function, limiting conditions must also be considered. Therefore, constrained and unconstrained optimization methods are closely associated together in view of the fact that the unconstrained optimization algorithms often form the basic idea for the constrained optimization methods.

Most problems which solve real system models involve nonlinear optimization with complicated objective functions or limiting conditions, where an analytical solution (for example by quadratic programming, geometric programming, etc.) is almost impossible. These cases lead a designer to use a different process, where at first the objective function is evaluated and afterwards it is gradually improved to achieve convergence. Generally, the nonlinear optimization procedure could be pictured as follows (Figure 6.1) [39].

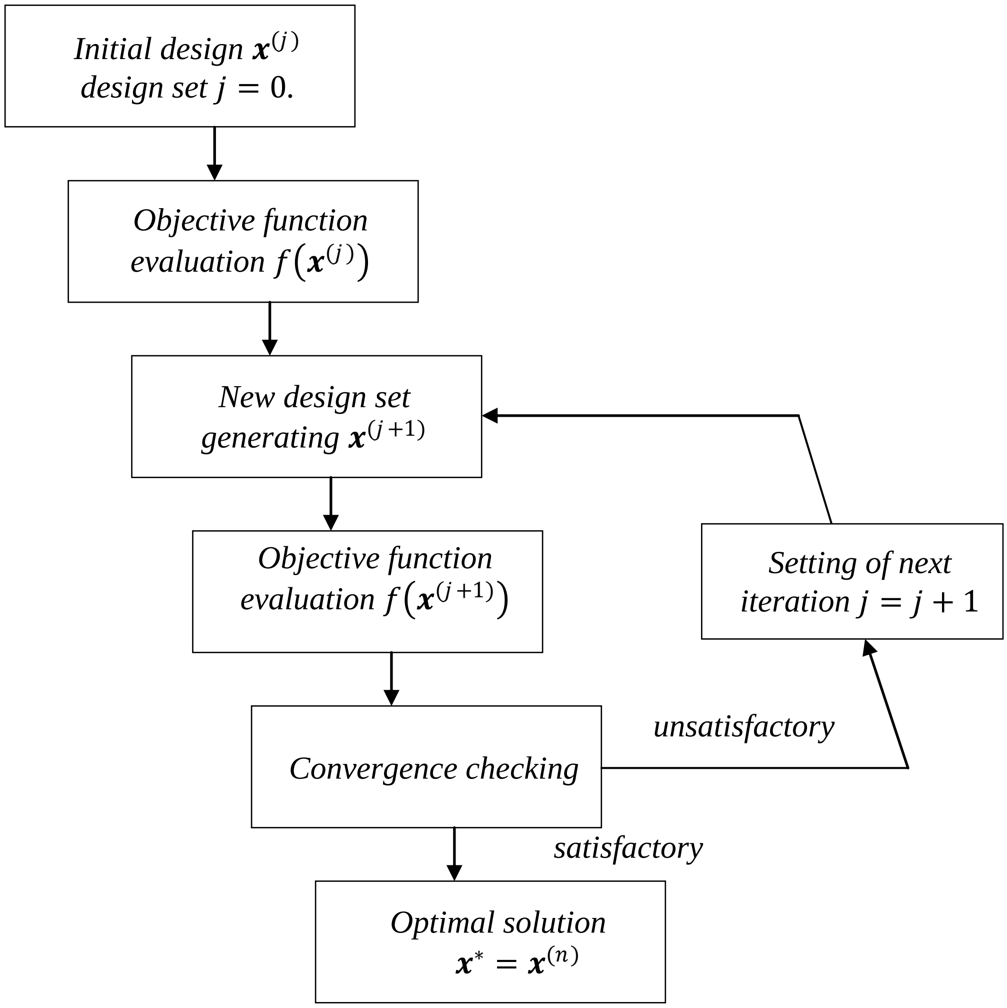

Figure 6.1 Nonlinear optimization process scheme

A general optimization problem definition of mathematical programming is defined as follows:

$$Find\,\boldsymbol{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}, which\ minimize\ f(\boldsymbol{x}) \qquad (6.2.1)$$

where $\boldsymbol{x}$ represents n-dimensional design vector and its members x_1 to x_n are design variables separately. With certain modifications, the minimum of the objective function f could be reached. It is possible to solve some specific cases manually, but most practical systems have to be solved by iteration procedures. The iteration procedure is determined by consecutive calculation, which starts with an initial design of $\boldsymbol{x}^{(0)}$ and continues till $\boldsymbol{x}^{(*)}$ is achieved. The obtained vector $\boldsymbol{x}^{(*)}$ is called optimum if it achieves the minimum of the objective function f. Mathematical programming offers several methods for solving unconstrained optimization problems. These could be classified as *direct* and *indirect* (*descent or gradient*) methods. The direct methods do not need minimization/maximization problem partial derivations of functions, only objective function values. Hence, they are often named *non-gradient* methods, or *zero order* methods. In comparison with the gradient methods, they are generally used for solving less difficult problems with a low number of defined variables. The descent methods require, in addition to functional values, the first derivation and in some cases also second derivation of the objective function. With function derivation, there is more information obtained about a problem. For this reason the gradient methods are considered as more accurate, but with a prejudice towards computing time. A list of some gradient and non-gradient methods is given in the following table (Table 6.1) [55]:

Table 6.1 Unconstrained optimization methods

Direct methods	Indirect methods
Random search method	*Steepest descent method*
Grid search method	*Conjugate gradient method*
Simplex method	*Newton's method*
Powell's method	*Quasi-Newton method*
Interpolation methods	*Marquardt method*
Statistical methods	
Approximate methods	

6.2.1 Direct (Approximation) Methods

Let’s consider the optimization process where the objective function is not defined, rather the problem is formulated by stated points. Then, the solution consists in substitution of one function by another, which is more suitable to work with [58]. The substitution can be performed by function interpolation, where it is required that the new function crosses all defined points. The disadvantage of this method is in the case where one or more points are defined incorrectly and the errors are projected to the interpolated function too. Then, the optimization procedure could be led to an incorrect result, or more often to a divergence of the optimization problem. Hence, approximation methods appear to be more suitable, because the aim is to substitute an original problem by approximated function, which does not have to cross the defined points accurately. The approximation method in comparison to the interpolation methods leads to a smoother function which results in better convergence. Even so, the smoother function might tend to neglect a global extreme of the objective function. However, the points substitution by approximation leads an optimization problem to find the extreme of approximated function. Softwares which have implemented an optimization algorithm to be helpful by creation of more efficient technical problems usually use approximation methods to substitute the original objective function from the point of view of their universality. It prevents difficult selection of a optimization method for wide scale of real technical, military or even liberal problems. One of the most popular approximation methods in theoretical and practical problems is the least square method.

6.2.1.1 Least Square Method

As it has been mentioned in previous chapters, the design space is bounded by design and state variables limits, which directly enters to the efficient design of a problem with use of the optimization algorithm. Substitution of random values to the design variables is not a guarantee that the objective function takes a feasible solution. An infeasible solution could be evoked by violation of other conditions of the problem, as limits of state variables. One of the methods applied in this work is the Subproblem Approximation Method, which is based on the objective function approximation in dependence on defined points. A location of the points in n-dimensional space can be obtained by using one of the optimization tools, for example Random Tool, or Sweep Tool (sections 7.3.1 and 7.3.2). The objective function approximation is performed by the least square method.

The least square method substitutes defined points by a function, with differences of square summations instead of absolute values of distances, because then it is still possible to consider distances as continuous differentiable values individually. A certain disadvantage could be a point which lies far away from the others, because its value might influence a function progression in the point location. In practical problems are mostly minimized vertical distances of the points from the curve (polynomial, surface, hyper polynomial), instead of normal distances, because there is a possibility to compute values y when values x are known in dependence on the certain function. An analytical expression of analysed points is also more suitable (simpler) than in a case with measuring of normal distances. With the vertical case it is also easier to generalize the process from solution, where the objective function is a linear function to a solution, and where the objective function is expressed by a polynomial, which depending on its dimension minimizes errors. The errors are brought about by deviations of the approximated function from the original points. There are three types of the least square fitting in the Subproblem Approximation method, which can be used. The first is a linear fitting, where the approximation function is expressed by linear line, the second is a quadratic fitting with a polynomial curve and the third is a fully quadratic expression with cross terms [7], [47], [75].

Let's assume the least square fitting is used to approximate a set of points by a linear function. Then the procedure is as follows:

The linear function is expressed by:

$$f = a + bx \qquad (6.2.2)$$

and vertical distance of any point from the line is:

$$e = |a + bx - f|. \qquad (6.2.3)$$

The aim of the least square method is to minimize a function:

$$E(a,b) = \sum_{i=1}^{n}(b_i x_i + a - f_i)^2 \qquad (6.2.4)$$

where n defines a number of the points, which are approximated by the linear function (6.2.2).

A local extreme in the differentiable function can occur at a stationary point,

$$\frac{\partial E}{\partial a} = 2(b_i \sum_{i=1}^{n} x_i^2 + a \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i f_i) \qquad (6.2.5)$$

and

$$\frac{\partial E}{\partial b} = 2(b_i \sum_{i=1}^{n} x_i + a \sum_{i=1}^{n} 1 - \sum_{i=1}^{n} f_i). \qquad (6.2.6)$$

Then, local extremes of the objective function are defined:

$$b_i \sum_{i=1}^{n} x_i^2 + a \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i f_i \qquad (6.2.7)$$

and

$$b_i \sum_{i=1}^{n} x_i + an = \sum_{i=1}^{n} f_i. \qquad (6.2.8)$$

Now, the approximation of the original points by a general polynomial function is presented.

Let the polynomial be in the form:

$$f_m(x) = a + a_1 x + a_2 x^2 + \cdots + a_m x^m = \sum_{j=0}^{m} a_j x^j . \qquad (6.2.9)$$

In most cases, an order of the polynomial is essentially lower than the number of points which are subjected to the approximation. If the number of the points was equal to the polynomial order, then the curve would cross the original points in a minimization of errors and the problem would be determined as interpolation. The least square method is based on a minimization of a squared errors summation:

$$E = \sum_{i=0}^{n} e_i^2 = \sum_{i=0}^{n} [p_n(x_i) - f(x_i)]^2 = \sum_{i=0}^{n} \sum_{j=0}^{m} \left[a_j x_i^j - f(x_i)\right]^2 . \qquad (6.2.10)$$

The first derivation has to be executable to reach the minimum of the function (6.1.10). So:

$$\frac{\partial E}{\partial a_k} = 0. \qquad (6.2.11)$$

By substitution and adaptation we obtain:

$$\frac{\partial E}{\partial a_k} = 2(\sum_{j=0}^{m} a_j \sum_{i=0}^{n} x_i^{j+k} - \sum_{i=0}^{n} f(x_i) x_i^k) = 0. \qquad (6.2.12)$$

From the point of view of a synoptic expression we take:

$$S_k = \sum_{i=0}^{n} x_i^m \qquad (6.2.13)$$

$$T_k = \sum_{i=0}^{n} f(x_i) x_i^k \qquad (6.2.14)$$

and then:

$$\sum_{j=0}^{m} a_j S_{j+k} - T_k = 0 \qquad (6.2.15)$$

which is an expression for a system of linear equations, wherein the polynomial coefficient values a_k are obtained. If some components in the least square problem solution are more important than the others, a weight function is established. The weights tend to emphasize or suppress a variable meaning within the solution. Then the equation (6.2.10) is reformulated as follows:

$$E = \sum_{i=0}^{n} \phi_i e_i^2 = \sum_{i=0}^{n} \phi_i [p_n(x_i) - f(x_i)]^2 = \sum_{i=0}^{n} \sum_{j=0}^{m} \phi_i \left[a_j x_i^j - f(x_i)\right]^2 \qquad (6.2.16)$$

where ϕ is a weight which is incorporated to a variable i.

6.2.2 Indirect (Descent) Methods

The indirect methods are based on derivation of an objective function. The aim is to find a gradient of the objective function, which is a vector whose members are based on partial derivations of a scalar field according to coordinates of the system. The objective function gradient has a general form in a n-dimensional field as follows:

$$\nabla f = \begin{Bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \\ \vdots \\ \dfrac{\partial f}{\partial x_n} \end{Bmatrix}. \qquad (6.2.17)$$

The most important property of the gradient ∇ in minimization/maximization of the objective function is that if the function moves in the direction of the gradient, than the functional value declines/increases the steepest possible way. Then, if the gradient presents the steepest descent, its negative value presents the steepest increasing of the function. So, the search procedure of the minimum or maximum of a function consists of the same process with opposite sign of the objective function gradient. The following text is directed at the *Steepest descent method* and the C*onjugate gradient method*.

6.2.2.1 Steepest Descent Method

The Steepest descent method proceeds by an iteration process from the initial design (initial point) in the direction of steepest descent till the desired minimum/maximum is achieved. Its algorithm is possible to summarize in the following steps [55]:

1. The first step is a determination of the initial point $\boldsymbol{x}^{(0)}$, which presents the initial design of a problem.
2. Searching of a direction $\boldsymbol{d}^{(j)}$ is the second step of the procedure, where

$$\boldsymbol{d}^{(j)} = -\nabla f_i = -\nabla f(\boldsymbol{x}_i). \qquad (6.2.18)$$

3. The third step consists of the setting of a suitable step length s_j in a direction $\boldsymbol{d}^{(j)}$ and the following setup:

$$\boldsymbol{x}^{(j+1)} = \boldsymbol{x}^{(j)} + s^{(j)}\boldsymbol{d}^{(j)}. \tag{6.2.19}$$

4. A checking of the new obtained point $\boldsymbol{x}^{(j+1)}$ from the optimality point of view, that is verifying if the process goes the right way (direction). In the case that the optimum $\boldsymbol{x}^*$ has been reached the calculation procedure is terminated. On the other hand, if the optimum $\boldsymbol{x}^*$ has not been reached yet, the process continuous with step 5.
5. A configuration of a new iteration $j = j + 1$ and the procedure continuous by repetition of the step 2.

The steepest descent method seems to be the most efficient optimization method in the unconstrained optimization problem field, because it progresses the fastest way to the extreme of the objective function. It stands to reason that the method has local qualities only. So, if the final function is not convex or concave only, or the initial point is not on a rising or decreasing curve around the global extreme, the method converges to the local extreme. The steepest descent method does not guarantee robust solution. Another disadvantage could be a slow convergence for long convex or concave functions, considering the many small steps.

6.2.2.2 Conjugate Gradient Method

The conjugate gradient method is commonly used for solving big and sparse systems of linear equations with symmetric and positive-definite matrices. It solves some characteristics where the steepest descent method fails. It uses the first derivation of the function $f(\boldsymbol{x})$ only. The main point of the method is that for a movement from a point $\boldsymbol{x}^{(j)}$ to point $\boldsymbol{x}^{(j+1)}$, it uses not only a new direction $\boldsymbol{d}^{(j+1)}$, but also the direction from the previous iteration $\boldsymbol{d}^{(j)}$. The reason is to receive information from both the actual and previous directions of the objective function. This information then allows us to reach a minimum or maximum of the function faster. The conjugate gradient method is more efficient with better convergence in functions whose shape is softly curved. Now, let's consider a quadratic function in a matrix form [55]:

$$f(\boldsymbol{X}) = \frac{1}{2}\boldsymbol{X}^T[\boldsymbol{H}]\boldsymbol{X} + \boldsymbol{B}^T\boldsymbol{X} + \boldsymbol{C} \tag{6.2.20}$$

where the matrix $[\boldsymbol{H}]$ is called Hessian matrix (section 5.2.2). At first, the objective function is minimized in $\boldsymbol{d}^{(0)} = -\nabla f\left(\boldsymbol{x}^{(0)}\right)$ direction (see steepest descent method described in section 6.2.2.1) with a certain step length $s^{(0)}$. The direction of the second searched vector $\boldsymbol{d}^{(1)}$ is given by the linear combination $\boldsymbol{d}^{(0)}$ and $-\nabla f(\boldsymbol{x}^{(1)})$:

$$\boldsymbol{d}^{(1)} = -\nabla f\left(\boldsymbol{x}^{(1)}\right) + \gamma^{(1)}\boldsymbol{d}^{(0)} \tag{6.2.21}$$

where the constant $\gamma^{(1)}$ can be determined by vectors association $\boldsymbol{d}^{(0)}$ and $\boldsymbol{d}^{(1)}$ considering the matrix $[\boldsymbol{A}]$.

$$\gamma^{(1)} = \frac{\left[\nabla f\left(\boldsymbol{x}^{(1)}\right) - \nabla f\left(\boldsymbol{x}^{(0)}\right)\right]^T \nabla f\left(\boldsymbol{x}^{(1)}\right)}{\nabla f(\boldsymbol{x}^{(0)}) \nabla f(\boldsymbol{x}^{(0)})} \tag{6.2.22}$$

From the previous definitions accrues a general formulation of the conjugate gradients, which corresponds to the Polak-Ribier definition for the jth searched direction:

$$\boldsymbol{d}^{(j)} = -\nabla f\left(\boldsymbol{x}^{(j)}\right) + \gamma^{(j)} \boldsymbol{d}^{(j-1)} \tag{6.2.23}$$

$$\gamma^{(j-1)} = \frac{\left[\nabla f\left(\boldsymbol{x}^{(j)}\right) - \nabla f\left(\boldsymbol{x}^{(j-1)}\right)\right]^T \nabla f\left(\boldsymbol{x}^{(j)}\right)}{\nabla f(\boldsymbol{x}^{(j-1)}) \nabla f(\boldsymbol{x}^{(j-1)})}. \tag{6.2.24}$$

The process of the minimization iteration computing performed by the conjugate gradient method is summarised in the following steps:

1. The initial point $\boldsymbol{x}^{(0)}$ is considered.
2. Determination of the direction vector $\boldsymbol{d}^{(0)} = -\nabla f(\boldsymbol{x}^{(0)})$, where $\nabla f(\boldsymbol{x})$ is the gradient of the objective function $f(\boldsymbol{x})$. The direction vector defines the search direction of the process.
3. Definition of point $\boldsymbol{x}^{(1)}$, which corresponds to expression:

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + s^{(0)^*} \boldsymbol{d}^{(0)} \tag{6.2.25}$$

   where $s^{(0)^*}$ is the suitable step length in the $\boldsymbol{d}^{(0)}$ direction. Another iteration follows by adjusting $j = 1$.
4. Set $\nabla f^{(\mathrm{j})} = \nabla f\left(\boldsymbol{x}^{(j)}\right)$ and evaluation of:

$$\boldsymbol{d}^{(j)} = -\nabla f\left(\boldsymbol{x}^{(j)}\right) + \frac{\left[\nabla f\left(\boldsymbol{x}^{(j)}\right) - \nabla f\left(\boldsymbol{x}^{(j-1)}\right)\right]^T \nabla f\left(\boldsymbol{x}^{(j)}\right)}{\nabla f(\boldsymbol{x}^{(j-1)}) \nabla f(\boldsymbol{x}^{(j-1)})} \boldsymbol{d}^{(j-1)} \tag{6.2.26}$$

5. The evaluation of the suitable step length $s^{(0)^*}$ in the $\boldsymbol{d}^{(j)}$ direction and determination of a new point

$$\boldsymbol{x}^{(j+1)} = \boldsymbol{x}^{(j)} + s^{(j)^*} \boldsymbol{d}^{(j)} \tag{6.2.27}$$

6. The procedure is repeated till convergence is achieved. If the point $\boldsymbol{x}^{(j+1)}$ is the desired extreme of the objective function, the process is terminated. If the optimum is not achieved, the procedure continues with step 4 by a configuration of $j = j + 1$ till the convergence criteria are satisfied.

6.2.3 Elimination Methods

In the case that an objective function is defined by one variable, a one-dimensional optimization problem must be solved. Elimination methods are used to find the maximum or minimum of unimodal functions, and can also be applied in minimization or maximization problems of even discontinuous functions. In this section the golden section method is discussed.

6.2.3.1 Golden Section Method

The last step to complete the previous gradient methods of the unconstrained optimization problem is to determine a suitable (optimal) step length s_j, which establishes a distance of the computational processing in a direction of the gradient $\boldsymbol{d}^{(j)}$. There are many methods, which can be used to determine the length of the step (for example see [55]). One of them is the golden section method. In the gradient search method processes criteria for a minimum and a maximum step length value are determined:

$$\underline{s_j} \leq s_j \leq \overline{s_j} \tag{6.2.28}$$

where $\underline{s_j} = 0$. It follows that the corners of the abscissa are defined as $\underline{s_j}$ and $\overline{s_j}$ and the aim is to find a point s_j between them. The point is often referred to as the optimal step length.

If the abscissa with length s is divided between points $\underline{s_j}$ and $\overline{s_j}$ to two sections $s_{(1)}$ and $s_{(2)}$, and after that the abscissa $s_{(1)}$ to $s_{(2)}$ and $s_{(3)}$ (Figure 6.2), then the correlation could be formulated as:

$$\frac{s_{(2)}}{s_{(1)}} = \frac{s_{(3)}}{s_{(2)}} \qquad \textit{and} \qquad s_{(3)} = s_{(1)} - s_{(2)}. \tag{6.2.29}$$

After substitution and conversion is obtained:

$$\frac{s_{(2)}}{s_{(1)}} = \frac{1 - \frac{s_{(2)}}{s_{(1)}}}{\frac{s_{(2)}}{s_{(1)}}}. \tag{6.2.30}$$

If $\frac{s_{(2)}}{s_{(1)}} = r$, then $r = \frac{1-r}{r}$ and with a substitution the following equation is given

$$r^2 + r - 1 = 0. \tag{6.2.31}$$

The positive solution of the quadratic equation (6.2.31) is:

$$r = \frac{\sqrt{5} - 1}{2}. \tag{6.2.32}$$

The ratio expressed by equation (6.2.32) is called the *gold section*.

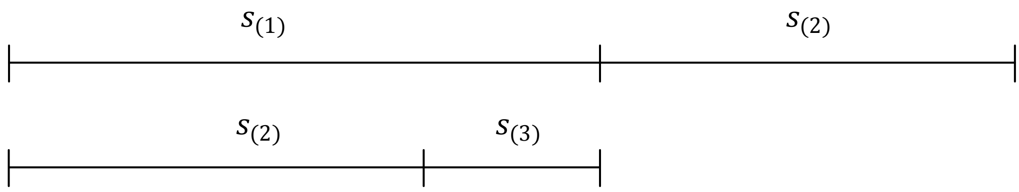

Figure 6.2 Golden Section Method

The obtained subranges are related in the follows ratios:

$$\frac{s_{(2)}}{s}:\frac{s_{(3)}}{s}:\frac{s_{(2)}}{s} \rightarrow \frac{3-\sqrt{5}}{2}:\sqrt{5}-2:\frac{3-\sqrt{5}}{2} \approx 0{,}38; 0{,}24; 0{,}38 \qquad (6.2.33)$$

By these methods the four coordinate values for independence variables (x_1, x_2, x_3, x_4) and their matching functional values (y_1, y_2, y_3, y_4) are obtained. From them the one chosen is the one which reaches the minimum values $y\left(x^{(min)}\right) = \min_k\left\{y\left(x^{(k)}\right)\right\}$. The bigger one of the two neighbouring intervals in whose centre is the point $x^{(\min)}$ is divided again by the gold section method. The process is repeated until the interval size is smaller than the allowable error tolerance.

6.3 CONSTRAINED OPTIMIZATION

The previous text was applied to some optimization algorithms, which solve an optimization case to search the minimum or maximum of a problem, where there are no constraints (6.2.1). But, in civil engineering structures or structural element designs some set of conditions is almost always presented, and must be considered. If such a problem is subjected to an efficient design using an optimization algorithm, the unconstrained optimization problem is not sufficient. In this case, the problem is pointing to the constrained optimization problem. Outcomes of the constrained optimization problem have to be always feasible (section 5.3.13). This means that the objective function reaches the minimum/maximum value and at the same time all determined criteria are satisfied. A general definition of the constrained optimization problem comes by extension from the general formulation of the unconstrained optimization problem (6.2.1.). Then, the constrained optimization problem is expressed as follows:

$$Find\,\boldsymbol{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}, which\ minimize\ f(\boldsymbol{x}) \tag{6.3.1}$$

subjected to:

$$\begin{aligned} g_i(\boldsymbol{x}) &\leq 0 \qquad && (i = 1,2,3, \dots, m) \\ l_i(\boldsymbol{x}) &= 0 \qquad && (i = 1,2,3, \dots, p) \end{aligned} \tag{6.3.2}$$

where $g_i(\boldsymbol{x})$ and $l_i(\boldsymbol{x})$ represent constraints of a design, where the first is expressed by a system of inequations and the second demonstrates a system of equations. Hence, unconstrained optimization problems are categorized to constrained optimization problems with equality and inequality constraints or combination of the two. A number of variables is not connected by any means with a number of defined constraints m and/or p. Algorithms which solve constrained optimization problems are generally considered as less efficient, because excepting an extreme search in terms of a function, it is necessary to think about the constraints. This led developers to employ the advantageous characteristics of unconstrained optimization problems. For this reason many constrained optimization methods, are based on the methods which solve the unconstrained optimization problems.

6.3.1 Design Constraints

The aim of the design optimization problem is to find the optimum value of an objective function, subjected to constraints, which can be expressed by system of equations and/or inequations. The constraints considerably affect properties of the whole mathematical programming procedure. As a consequence of the optimization problem constraints, the following situations can occur:

1. The constraints of the optimization problem do not affect the searched extreme of the objective function. This means that the demanded optimum and the minimum (maximum) of the objective function are the same and they are in the space of feasible solutions (see Figure 6.3). Although the constraints are defined, the problem can be solved by an unconstrained optimization method. However, in practical cases, where the real system is transformed to a certain mathematical formulation, it is rarely possible to establish whether the optimum is affected by constraints or not. So even the optimization problem would be simpler to solve without considering the constraints. It is recommended to use one of the constrained optimization problem method and take constraints into account during the optimization process.

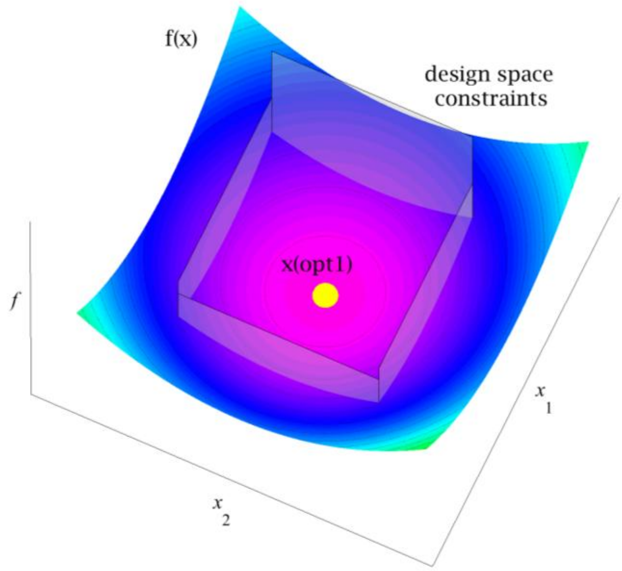

Figure 6.3 Constraints without influencing the minimum

2. The second situation is the optimum point location on the border of a design constraint (Figure 6.4). This usually occurs if one of the defined conditions achieves the limit value of some physical property of a problem. In civil engineering problems it could be, for example, to achieve maximum value of a displacement of an element, maximum strain, etc.

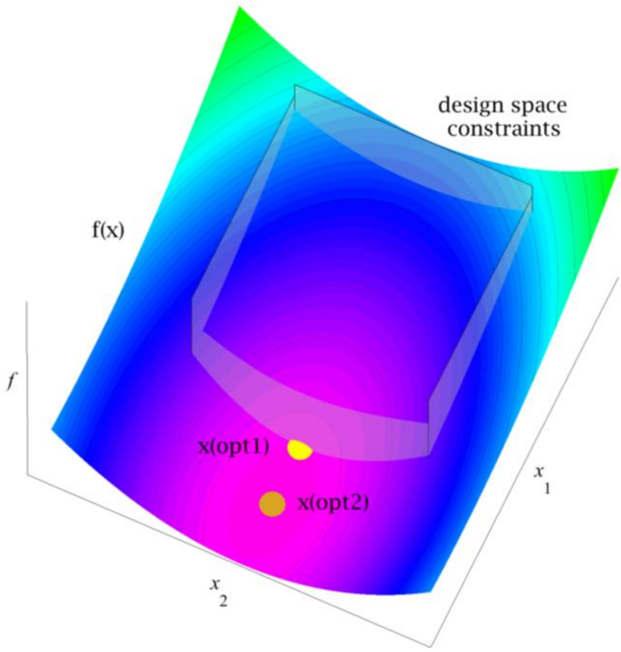

Figure 6.4 Minimum is located on the border of feasible design space, global minimum is out of feasible design space

3. If the objective function has more than one unconstrained minimum/maximum point, when design constraints are applied, it is possible that the constrained optimization problem can achieve more than one optimal feasible solution (Figure 6.5).

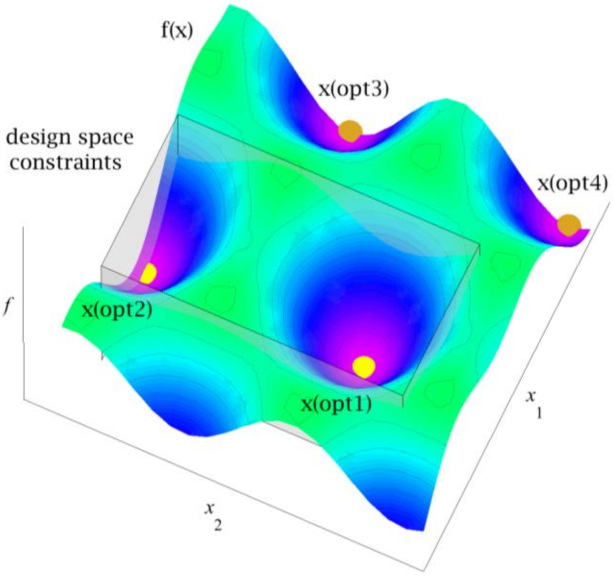

Figure 6.5 Two minimum points in feasible design space

4. The objective function has only one minimum/maximum in the unconstrained problem. But if the design constraints are considered, more than one optimum point is achieved (Figure 6.6).

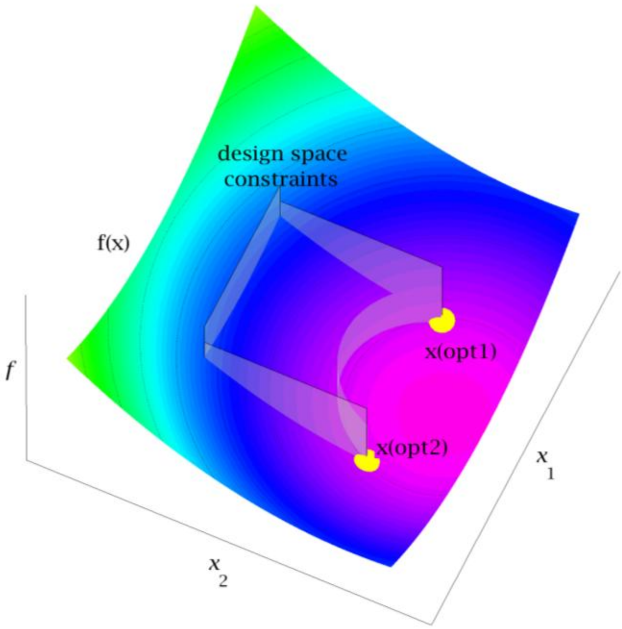

Figure 6.6 Two minimum points on constraints' borders

6.3.2 Penalty Function

Many constrained optimization methods are based on algorithms which solve unconstrained optimization problems, therefore the constraints have to be considered in the design. The consideration is usually performed by penalty functions, whose expressions are added to the unconstrained function. The penalty functions compensate for the demanded design constraints.

The penalty function method transforms the constrained optimization problem into the unconstrained optimization method. This is achieved by substitution of the defined design conditions with penalties, which are considered as adjuncts in functions of dependent variables (objective function and state variables). If limit values of variables are violated, the solution is considered infeasible according to the penalty function value. Consider a general constrained optimization problem defined as follows:

$$Find\,\boldsymbol{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix},\ which\ minimize\ f(\boldsymbol{x}) \tag{6.3.3}$$

subjected to the constraints, which are expressed by a system of inequations:

$$g_i(\boldsymbol{x}) \leq 0 \qquad\qquad (i = 1,2,3,\dots,m). \tag{6.3.4}$$

Furthermore, this problem is transformed to the unconstrained optimization problem through the penalty function application in the form:

$$f(\boldsymbol{x}, q_k) = f(\boldsymbol{x}) + q^{(j)} \sum_{i=1}^{m} P_i[g_i(\boldsymbol{x})] \tag{6.3.5}$$

where P_i is the penalty function according to the constraint formulation $g_i(\boldsymbol{x})$ and q is a penalty parameter. If the unconstrained minimization problem (6.2.1) is performed by an iterative process with consideration of the penalty parameters in each iteration $q^{(j)}$ $(k = 1,2,3,\dots,n)$, the optimization procedure progresses identically with the unconstrained optimization problem (6.2.1). Hence the penalty function method is considered as one of the sequential unconstrained minimization techniques (SUMT). The penalty function methods are threefold: interior, exterior and extended interior. For example, a general expression of the interior penalty function P_i [55] is given as:

$$P_i = -\frac{1}{g_i(\boldsymbol{x})} \tag{6.3.6}$$

and the exterior penalty function has the form:

$$P_i = max[0, g_i(\boldsymbol{x})]. \tag{6.3.7}$$

If the interior penalty function is applied, all optimization function values of an unconstrained optimization problem $f^{(j)}$, in each iteration j, occur inside the feasible space (section 5.3.13) of an original constrained optimization problem. This is achieved by an appropriate penalty function $q^{(j)}$ modification in a defined range. In this case convergence is achieved by a sequential detraction of the penalty parameter value $q^{(j)}$. Otherwise, if the external penalty function is used, the objective function values are in the infeasible space, and simultaneously with the convergence are getting nearer to the optimal point. In this case, the penalty parameter $q^{(j)}$ increases its value till the minimum objective function value is achieved.

6.3.2.1 Exterior Penalty Function

A general formulation of the exterior penalty function f can be expressed as follows:

$$f\left(\boldsymbol{x}, q^{(j)}\right) = f(\boldsymbol{x}) + q^{(j)} \sum_{i=1}^{m} [g_i(\boldsymbol{x})]^{\lambda} \qquad (6.3.8)$$

where $q^{(j)}$ is a penalty parameter for certain design constraint, which is expressed by a function $g_i(\boldsymbol{x})$ and the term λ is a non-negative constant. The function $g_i(\boldsymbol{x})$ value is obtained from:

$$g_i(\boldsymbol{x}) = max\langle g_i(\boldsymbol{x}), 0\rangle = \begin{cases} g_i(\boldsymbol{x}) & if \quad g_i(\boldsymbol{x}) > 0 \quad {\scriptstyle (1)} \\ 0 & \quad g_i(\boldsymbol{x}) \le 0 \quad {\scriptstyle (2)} \end{cases}. \qquad (6.3.9)$$

(1) The design constraints are violated.

(2) The objective function reaches feasible values within the frame of defined constraints $g_i(\boldsymbol{x})$.

From the equation (6.3.8) it is obvious that the penalty parameter $q^{(j)} \sum_{i=1}^{m} [g_i(\boldsymbol{x})]^{\lambda}$ markedly increases the objective function value $f(\boldsymbol{x}, q^{(j)})$ if the design constraints $g_i(\boldsymbol{x})$ are violated by the parameter λ. In view of the fact that the aim of the optimization process is to reach a minimum or maximum value of the objective function $f\left(\boldsymbol{x}, q^{(j)}\right)$, any obtained result sets which exceed the assumed objective function values too much are suspended (penalized) from the solution. The minimum functional values $f\left(\boldsymbol{x}, q^{(j)}\right)$ are usual throughout the whole optimization procedure in infeasible design space. The process converges to the optimum $\boldsymbol{x}^{(j)^*}$ simultaneously with $j \to \infty$ and $q^{(j)} \to \infty$. Thus, the minimum values in each iteration obtained are slowly getting nearer to the defined design space boundaries simultaneously with $j \to \infty$ till the $\boldsymbol{x}^{(j)^*}$ is achieved, which finally satisfies all determined conditions of the optimization problem, and it is then a component of the feasible design set in the constrained optimization problem. Effects of the λ value changes can be described by the following points:

1. $\lambda = 0$. So, the function (6.3.8) is in the form:

$$f(\boldsymbol{x}, q^{(j)}) = f(\boldsymbol{x}) + q^{(j)} \sum_{i=1}^{m} [g_i(\boldsymbol{x})]^0 = \begin{cases} f(x) + mq^{(j)} & \text{if} \quad g_i(x) > 0 \\ f(x) & \quad\quad g_i(x) \leq 0 \end{cases}. \tag{6.3.10}$$

   This function is discontinuous on the feasible space boundaries. A minimization of the problem would lead to very complicated computation during the whole optimization procedure.
2. $0 < q < 1$. In this case the function $f(\boldsymbol{x}, q^{(j)})$ is continuous. The discontinuity occurs when the first derivation is performed along the boundaries, which define the feasible design space. In this case, during the minimization process of the function $f(\boldsymbol{x}, q^{(j)})$ the complications occur again. The penalty parameter $q^{(j)} \sum_{i=1}^{m} [g_i(\mathbf{x})]^{\lambda}$ reaches small numbers at this point. If a very small number of the penalty parameter is achieved the result set could be considered as feasible, although in reality it is not correct.
3. $q = 1$. It is a similar situation to the previous one. Thus, the discontinuity of the first derivation of $f(\boldsymbol{x}, q^{(j)})$ along the boundaries of the feasible design space occurs. With this, the exterior penalty function method loses efficiency, especially from the computing point of view [55].
4. $q > 1$. The function $f(\boldsymbol{x}, q^{(j)})$ has continuous first derivations, which are given as:

$$\frac{\partial F}{\partial \boldsymbol{x}_i} = \frac{\partial f}{\partial \boldsymbol{x}_i} + q^{(j)} \sum_{i=1}^{m} \lambda [g_i(\boldsymbol{x})]^{\lambda-1} \frac{\partial g_i(\boldsymbol{x})}{\partial \boldsymbol{x}_i}. \tag{6.3.11}$$

   Most practical problems consider $\lambda = 2$.

The procedure of the exterior penalty function method can be summarized into the following points:

1. The start consists of defining of an arbitrary point $\boldsymbol{x}_0$ and a suitable penalty parameter number q_0.
2. The second step is to find a new vector $\boldsymbol{x}_1$, which reduces the objective function value:

$$F(\boldsymbol{x}, q^{(j)}) = f(\boldsymbol{x}) + q^{(j)} \sum_{i=1}^{m} [g_i(\boldsymbol{x})]^{\lambda}. \tag{6.3.12}$$

3. Checking of feasibility of the obtained solution $\boldsymbol{x}^{(j)}$. If the vector $\boldsymbol{x}^{(j)}$ satisfies all the design conditions and limits (it is in the feasible design space), it represents the optimum of the solution, so $\boldsymbol{x}^{(j)} = \boldsymbol{x}^{(j)^*}$. If the obtained point is not the optimum, the procedure continues with the following step.

The fourth step consists of the definition of a new penalty parameter value which satisfies $q^{(j+1)} > q^{(j)}$. The $q^{(j+1)}$ value is usually chosen so as to correspond to the equation $q^{(j+1)} = cq^{(j)}$, where c is a constant whose value is much bigger than 1.

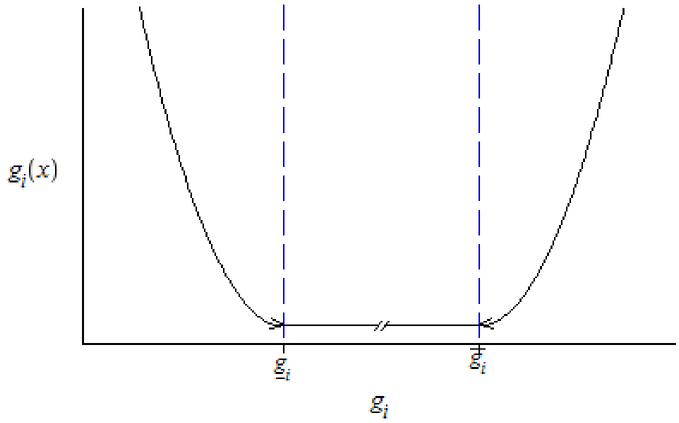

Figure 6.7 Exterior penalty function

6.3.2.2 Interior Penalty Function

As was mentioned previously, the aim of the penalty function method is to transform the constrained optimization problem to the unconstrained optimization problem by adding penalty terms to the objective function. The interior penalty function method, as compared to the exterior penalty function method performs the solution inside of the feasible design space. If the optimization procedure goes further from the feasible design space boundaries, the penalty parameter declines and if it is to be to the contrary (the process goes toward to the feasible design space), the penalty parameter value increases. The influence of the penalty function parameter is also changing the objective function value. Thus, by getting nearer to the feasible design space boundaries, the penalty parameter reaches very high values, therefore the substantial increment of the objective function value is the cause of the penalty parameter. If the boundaries are achieved, the penalty parameter value leads to infinity. A general objective function F form, extended by the penalty parameter term, is as follows:

$$F\left(\boldsymbol{x}, q^{(j)}\right) = f(\boldsymbol{x}) - q^{(j)} \sum_{i=1}^{m} \frac{1}{g_i(\boldsymbol{x})}. \tag{6.3.13}$$

It is obvious from the equation (6.3.13), that if the $g_i(\boldsymbol{x})$ value is negative (it satisfies all the design conditions (6.3.4)), the objective function value F is bigger than f. If the feasible design boundaries are achieved, so $g_i = 0$, the objective function value F leads to infinity. The penalty function term in the equation (6.3.13) cannot be defined in the infeasible space. This means that the initial design point $\boldsymbol{x}^{(0)}$ has to be always a point which satisfies all the determined conditions of the problem, thus being inside of the feasible design space. Then:

$$g_i(\boldsymbol{x}^0) < 0 \qquad\qquad (i = 1,2,3, \dots, m). \tag{6.3.14}$$

The iteration process of the objective function minimization using the interior penalty function method is as follows:

1. The first step is to select an initial point $\boldsymbol{x}^{(0)}$, which is necessarily inside of the feasible design space (6.3.14) and the choice of the initial penalty parameter $q^{(0)} > 1$.
2. The second step is a minimization of the objective function $F(\boldsymbol{x}, q^{(j)})$ using any method which solves the unconstrained optimization problems and obtains the optimal design $\boldsymbol{x}^{(j)^*}$.
3. Check the obtained design $\boldsymbol{x}^{(j)^*}$. If $\boldsymbol{x}^{(j)^*}$ is the optimal design which satisfies all conditions of the problem, the optimization procedure is terminated. If not, the process continues as follows.
4. Definition of next penalty parameter $q^{(j+1)}$ so, that

$$q^{(j+1)} = cq^{(j)} \tag{6.3.15}$$

   where c corresponds to $c < 1$.
5. In the next iteration we set $j = j + 1$ and define a new initial point as $\boldsymbol{x}^{(0)} = \boldsymbol{x}^{(j)^*}$. The following procedure repeats points 2 to 5.

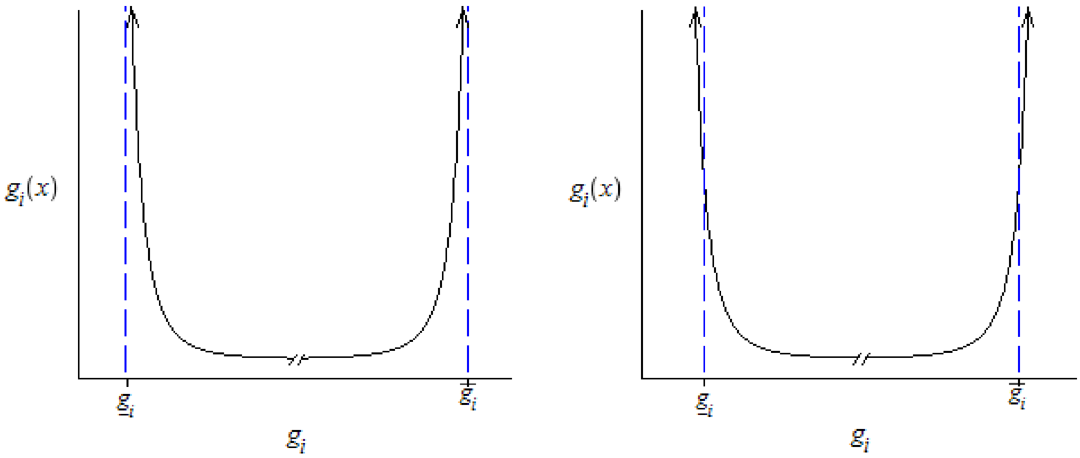

Figure 6.8 Interior and extended interior penalty function

6.3.2.3 Extended Interior Penalty Function

If the optimization process is performed by some one-dimensional optimization method (for example the golden section method described in section 6.2.3.1), it might happen that such a length of the step s_j is found, which leads the process to the infeasible design space. In this case, it is not possible to use the interior penalty function method (section 6.3.2.2). Hence an alternative method, *extended interior penalty function*, is used. The method makes solving of problems where the function F is defined in the infeasible design space possible. It combines the best properties of the interior and exterior penalty function methods in the constrained optimization problems, where the conditions are prescribed by a system of inequations.

If the extended interior penalty function is applied, the function F could be defined as follows:

$$F\left(\boldsymbol{x}, q^{(j)}\right) = f(\boldsymbol{x}) + q^{(j)} \sum_{i=1}^{m} g_i(\boldsymbol{x}) \qquad (6.3.16)$$

where

$$g_i(\boldsymbol{x}) = \begin{cases} -\dfrac{1}{g_i(\boldsymbol{x})} & \text{if} \quad g_i(\boldsymbol{x}) \le \varepsilon \\ -\dfrac{2\varepsilon - g_i(\boldsymbol{x})}{\varepsilon^2} & \quad g_i(\boldsymbol{x}) > \varepsilon \end{cases}. \qquad (6.3.17)$$

ε is a constant, which is expressed by a small negative number. This number makes the extension of the interior penalty function from $g_i(\boldsymbol{x}) \le \varepsilon$ to $g_i(\boldsymbol{x}) > \varepsilon$ possible. Then, the objective function behaviour is more lucid, not only in the feasible design space, but also at close quarters. The constant number ε is chosen so that there are positive objective function values on the feasible design space boundaries obtained.

$$\varepsilon = -c(q^j)^a \qquad (6.3.18)$$

Where constant a is selected from an interval $\frac{1}{3} \le a \le \frac{1}{2}$, whereas the number $a = \frac{1}{3}$ guarantees a raising of the penalties for violated conditions along with descending of $q^{(j)}$ to zero. The number $a = \frac{1}{2}$ helps to keep the minimal point $\boldsymbol{x}^{(j)^*}$ in a quadratic form of the penalty function. At the beginning of the optimization process the ε value chosen is in an interval $-0{,}3 \le \varepsilon \le -0{,}1$. The value $q^{(0)}$ is selected so that $f(\boldsymbol{x})$ and $q^{(0)} \sum_{i=1}^{m} g_i(\boldsymbol{x})$ are to be equal for the initial design $\boldsymbol{x}^{(0)}$. Thereby a value of the constant c is obtained in the equation (6.3.18). At the beginning of each iteration, the value ε is computed considering the actual $q^{(j)}$ from the equation (6.3.18) and stays constant for all its time.

6.3.3 Sequential Unconstrained Minimization Techniques (SUMTs)

The penalty function method becomes unstable and ineffective in cases where high accuracy for high $q^{(j)}$ number is required. This is incurred by rounding errors in the final function, which could be affected by an incorrect definition of the design set movement direction. Universal optimization methods, which are usually used for a broad range of optimization problems, can lead to these difficulties very often, with the result that the solution diverges or collapses. A suitable device in these situations could be to choose a sequential process, which is performed by sequential increasing of the penalty parameter $q^{(j)}$ till a limit of any defined optimization variable is achieved, then the sequential unconstrained minimization techniques (SUMTs) are applied [19], [23], [62]. This method leads the procedure to find an extreme of the original objective function $f(x)$. If the final function is convex, the solution is the global minimum. If in the feasible design space there is more than one local extreme, it is recommended to perform the whole procedure with different initial values again. The general process of the sequential unconstrained minimization technique can be described by the following steps:

1. The first step is to determine a convergence tolerance ε, start point $\boldsymbol{x}^{(0)}$ and initial penalty parameter $q^{(0)}$.
2. Furthermore, the objective function $f(\boldsymbol{x}, q)$ minimization is performed by any of the unconstrained optimization methods to find its extreme $x^*(q^{(j)})$.
3. The next step is to define a new penalty parameter:

$$q^{(j+1)} = cq^{(j)} \tag{6.3.19}$$

where the constant c is an arbitrary positive number.

7 DESIGN OPTIMIZATION METHODS AND TOOLS

The rapid development in information technology and systems which are designed for simulation of practical problems supports the use of optimization techniques for their design. The mathematical algorithms which are established to find the minimal or maximal values of a function require wide skills in mathematics and operating research. An optimization is a difficult mathematical process to find the minimum or maximum of an objective function which is mostly based on iterative procedure. If the aim of a project is to find an efficient design using any of the available optimization technique manually, it is necessary to create as simple a mathematical model as possible. This leads a designer to use general coefficients to guarantee the safety of a design and the design ends up moving away from reality. For that reason specialists for operating research and systems of information technologies started to deal with implementation of optimization techniques in mechanical and civil engineering softwares. Then, the designers have the opportunity to apply them for the creation of an efficient design. One way of using optimization methods in structural designs is to implement them in FEM- (Finite Element Method) based softwares. FEM is currently one of the most widely used methods in mechanical and civil engineering design. It is possible to categorize optimization techniques from many points of view, depending on the problem. For example, according to number of variables, number of objective functions, robustness of a design, linearity or nonlinearity of functions, presence of equal and unequal conditions, or their combination, etc. The next chapter deals with a characterization of optimization algorithms which are implemented in individual optimization module using the multi-physical finite element method program ANSYS. The module is established to find an efficient design of problems solved in the software. Two optimization methods are implemented in the module; First Order Method and Subproblem Approximation Method [75], [76]. Both methods transform constrained optimization problems to unconstrained optimization problems using penalty functions. Then, the chapter explains tools which could be helpful in solving optimization problems and making optimization methods more effective in finding a minimum value of an objective function.

7.1 SUBPROBLEM APPROXIMATION METHOD

The approximation method (SAM - Subproblem Approximation Method) is an iterative method based on an approximated function. Design variables are presented as independent variables, which are changing their values while the optimization is processing in each iteration till convergence is achieved. Each design variable is assigned upper and lower values to define constrained design. State variables are determined as dependent variables. They express function of design variables and their values are changing depending on varying members of a design vector. Their range is defined by lower and/or upper limits, depending on the problem. By limiting values of the design and state variables a feasible design space is defined. An objective function is a dependent function and the goal is its minimization or maximization within the frame of the feasible designs.

At first the approximation of the dependent variables (Obj - objective function and SVs - state variables) by least squares fitting is performed, and then the approximated objective function is minimized or maximized. Thus, the aim of the process is minimizing or maximizing an approximated function instead of the true function. As described in the previous sections (sections 6.1 and 6.2), more efficient methods for finding the extreme of a function are unconstrained optimization methods. For that reason, the defined constrained optimization problem is converted to an unconstrained optimization problem. The transformation is performed by the penalty function method, which is applied to the objective function. The penalty function replaces the previously defined constraints in limits of DVs and SVs.

7.1.1 Function Approximation

When a certain amount of iterations is executed, expressed by equations (7.1.1 - 7.1.4) below, with use of one or more optimization tools (section 7.3), the SAM technique could be described by the following steps:

1. Approximation of the dependent variables (Obj and SVs) by application of least squares fitting.
2. The constrained optimization problem is converted to the unconstrained optimization problem by the penalty function method.
3. The iteration process is performed.
4. Termination of the calculation.

The first step in the minimizing of the general constrained function

$$f = f(\boldsymbol{x}) \qquad (7.1.1)$$

subject to:

$$g_i(\boldsymbol{x}) \le \bar{g}_i \qquad (i = 1,2,3 \dots, m_1) \qquad (7.1.2)$$

$$\underline{h}_i \le h_i(\boldsymbol{x}) \qquad (i = 1,2,3 \dots, m_2) \qquad (7.1.3)$$

$$\underline{w}_i \le w_i(\boldsymbol{x}) \le \overline{w}_i \qquad (i = 1,2,3, \dots, m_3) \qquad (7.1.4)$$

is approximation every dependent variable (Obj and SVs).

The approximation of the dependent variables could be expressed by:

$$\hat{f}(\boldsymbol{x}) = f(\boldsymbol{x}) + error \tag{7.1.5}$$

$$\hat{g}(\boldsymbol{x}) = g(\boldsymbol{x}) + error \tag{7.1.6}$$

$$\hat{h}(\boldsymbol{x}) = h(\boldsymbol{x}) + error \tag{7.1.7}$$

$$\widehat{w}(\boldsymbol{x}) = w(\boldsymbol{x}) + error \tag{7.1.8}$$

where ˆ marks the approximated function. The optimization is an iterative process, that is why the general optimization problem has to contain not only number of variables i, but also quantity of performed loops j. The approximation of the objective function is performed in every step of the process until the convergence of the problem is achieved. The general approximated objective function is defined in a fully quadratic form with cross terms as:

$$\hat{f} = a + \sum_{i=1}^{n} b_i \boldsymbol{x}_i + \sum_{i=1}^{n} c_i x_i^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij}\, \boldsymbol{x}_i \boldsymbol{x}_j \tag{7.1.9}$$

where n presents number of performed loops. The form of each variable is done internally from iteration to iteration. To determine coefficients a, b_i, c_i and d_{ij}, the weighted least square technique is used. For example, the weighted least squares error norm for the objective function is in the form:

$$E^2 = \sum_{j=1}^{n_d} \phi^{(j)} \left(f^{(j)} - \hat{f}^{(j)}\right)^2 \tag{7.1.10}$$

where $\phi^{(j)}$ is the weight attached to the design set j and n_d is current number of design sets.

The errors norm for each SVs E^2 are formed similarly. The coefficient in the equation (7.1.10) are obtained by minimization of E^2. Varying weight coefficient values $\phi^{(j)}$ enables the designer to focus on any optimization variable throughout the entire optimization procedure. The more important variable, therefore, will have a higher weight coefficient. In the case where there are no privileged variables, weight coefficients tend to be the same. In such a case, the numbers of weight coefficients are set to 1 and the process consists of solving the simple least square method. The error norm is in the following form:

$$E^2 = \sum_{j=1}^{n_d} 1 \left(f^{(j)} - \hat{f}^{(j)}\right)^2 = \sum_{j=1}^{n_d} \left(f^{(j)} - \hat{f}^{(j)}\right)^2. \tag{7.1.11}$$

The weight coefficients are defined using one of the following methods:

1. The weight is pointing to the objective function value. For example, the higher weights have design sets with lower values of the objective function.
2. The weights are based on the design variables. The design sets which achieve better results have higher weight coefficient values than the others.
3. Based on feasibility of design sets. The feasible design sets have higher weights than the infeasible design sets.
4. A combination of previous points 1.-3.
5. All weights are unified. It means that: $\varphi^{(j)} = 1$, for all j.

The weights, which are defined to stress, or suppress the meaning of any variable can be in the range $\langle 0,1 \rangle$, where stress on the variable increases with higher weight coefficient values. With lower weight coefficient value, the suppress increases simultaneously during the optimization procedure.

A certain number of design sets must exist to approximate a dependent variable SVs or Obj. They can be performed by one of the optimization tools described in section 7.3. If any optimization tool is not used, a random design is performed by Random Tool (section 7.3.1) till required number of design sets is reached. This could be expressed as:

$$n_d < n + 2 \rightarrow \quad \textit{design sets are generated by random or other optimization tool}$$

$$n_d \geq n + 2 \rightarrow \quad \textit{the approximation is performed}$$

where:
n is a number of design variables
n_d is a number of design sets
With more design sets, the number of terms in equation (7.1.9) increases.

7.1.2 Minimizing of the Subproblem Optimization Problem

If the functions are approximated, then the minimizing problem is in the following form:

$$\hat{f} = \hat{f}(\boldsymbol{x}) \tag{7.1.12}$$

subject to:

$$\underline{\boldsymbol{x}}_i \leq \boldsymbol{x}_i \leq \overline{\boldsymbol{x}}_i \qquad (i = 1,2,3,\ldots,n) \tag{7.1.13}$$

$$\hat{g}_i(\boldsymbol{x}) \leq \overline{g}_i + \alpha_i \qquad (i = 1,2,3,\ldots,m_1) \tag{7.1.14}$$

$$\underline{h}_i - \beta_i \leq \hat{h}_i(\boldsymbol{x}) \qquad (i = 1,2,3,\ldots,m_2) \tag{7.1.15}$$

$$\underline{w}_i - \gamma_i \leq \hat{w}_i(\boldsymbol{x}) \leq \overline{w}_i + \gamma_i \qquad (i = 1,2,3,\ldots,m_3) \tag{7.1.16}$$

The next step of the procedure is to transform the constrained optimization problem to unconstrained. This is performed by penalty function. Then the minimizing problem with use of the Subproblem Approximation Method is expressed by:

$$\textit{Minimize} \qquad F\left(\boldsymbol{x}, q^{(j)}\right) = \hat{f} + f_0 q^{(j)} \left(\sum_{i=1}^{n} X(\boldsymbol{x}_i) + \sum_{i=1}^{m_1} G(\hat{g}_i) + \sum_{i=1}^{m_2} H(\hat{h}_i) + \sum_{i=1}^{m_3} W(\widehat{w}_i) \right) \tag{7.1.17}$$

where X is the penalty function, which is used to express the constraints of design variables, and G, H and W are penalty functions which substitute constraints of state variables. The reference objective function value f_0 is performed to reach consistent units. The unconstrained objective function $F\left(x, q^{(j)}\right)$ is changing its values simultaneously with the design variables and value $q^{(j)}$, which corresponds to penalty parameter. Solving of the equation (7.1.17) is performed by a sequential unconstrained minimization technique (SUMT) (section 6.3.3) in each iteration of the procedure. The superscript j presents using of subiterations which are performed during the optimization process, where the penalty parameter value $\left(q^{(j)} < q^{(j+1)} < q^{(j+2)}\, atd\right)$ increases gradually until the demanded convergence is achieved.

All the penalty functions are presented by the extended interior penalty function. For example, close to the upper limit of a design variable is the penalty function formed as:

$$X(\boldsymbol{x}_i) = \begin{Bmatrix} c_1 + \dfrac{c_2}{(\overline{x} - x_i)} & if \quad x_i < \overline{x} - \varepsilon\left(\overline{x} - \underline{x}\right) \\ c_3 + \dfrac{c_4}{(x_i - \overline{x})} & if \quad x_i \geq \overline{x} - \varepsilon\left(\overline{x} - \underline{x}\right) \end{Bmatrix} \qquad (i = 1,2,3,\dots,n) \tag{7.1.18}$$

where c_1, c_2, c_3, c_4 are internally evaluated constants and ε is very small positive number.

The penalty functions which substitute the state variables are in a similar form. For example, near to the upper limit again, it is formed:

$$W(w_1) = \begin{Bmatrix} d_1 + \dfrac{d_2}{(\overline{w} - \widehat{w}_i)} & if \quad \widehat{w}_i < \overline{w}_i - \varepsilon\left(\overline{w}_i - \underline{w_i}\right) \\ d_3 + \dfrac{d_4}{(\widehat{w} - \overline{w})} & if \quad \widehat{w}_i \geq \overline{w}_i - \varepsilon\left(\overline{w}_i - \underline{w_i}\right) \end{Bmatrix} \qquad (i = 1,2,3,\dots,m_1) \tag{7.1.19}$$

where d_1, d_2, d_3 and d_4 are internally evaluated constants. The similar expression is for G and H.

The sequential unconstrained minimization technique is used to reach the minimization of unconstrained objective function $F''^{(j)}$ at design iteration j. So:

$$\boldsymbol{x}^{(j)} \rightarrow \tilde{\boldsymbol{x}}^{j} \;\; \textit{along with}\, F^{(j)} \rightarrow \tilde{F}^{(j)} \tag{7.1.20}$$

where $\widetilde{\boldsymbol{x}}^{(j)}$ is the design variable vector, which correspond to $F''^{(j)}$.

The last step which is performed in each loop of the procedure is to create a design variable vector for following iteration $(j+1)$. Then the vector $\boldsymbol{x}^{(j+1)}$ is expressed as:

$$\boldsymbol{x}^{(j+1)} = \boldsymbol{x}^{(b)} + C\left(\widetilde{\boldsymbol{x}}^{(j)} - \boldsymbol{x}^{(b)}\right) \qquad (7.1.21)$$

where $\boldsymbol{x}^{(b)}$ are constants of the best sets that were performed and C is an internally chosen number in the range of $\langle 0,1 \rangle$. The value depends on the number of the infeasible sets.

7.1.3 Convergence

A termination of the optimization procedure is set, when convergence criteria are satisfied or the process is manually terminated. The convergence criteria are activated up to satisfy the equation (7.1.22), i.e. unless the current number of the design sets n_d is equal, or bigger than number of design sets, which are needed to form the approximations of the functions. The convergence is achieved, if an advance tolerance is satisfied. The tolerance is defined by differences of two in sequence objective function values or design variables in obtained design sets. The necessary conditions for convergence can be expressed as:

$$\left|f^{(j)} - f^{(j-1)}\right| \leq \tau \qquad (7.1.22)$$

$$\left|f^{(j)} - f^{(b)}\right| \leq \tau \qquad (7.1.23)$$

$$\left|x_i^{(j)} - x_i^{(j-1)}\right| \leq \rho_i \qquad (i = 1,2,3,\dots,n) \qquad (7.1.24)$$

$$\left|x_i^{(j)} - x_i^{(b)}\right| \leq \rho_i \qquad (i = 1,2,3,\dots,n) \qquad (7.1.25)$$

where τ is a tolerance defined as difference of two objective function values and ρ_i is a tolerance which is expressed by difference of design variables i. Both tolerances are defined before starting an optimization procedure. In the case that the tolerance criteria do not satisfy any of the defined equations (7.1.22 - 7.1.25) the solution diverges. This means that the convergence is not achieved in the defined design space and it is probably necessary to alter initial conditions of the problem. They could be performed by changing the initial variables or their limit values. For that reason, a designer defines maximum number of iterations N_s or maximum number of infeasible design sets N_{si} in sequence before the optimization procedure starts. Thus:

$$n_s = N_s \qquad (7.1.26)$$

$$n_{si} = N_{si} \qquad (7.1.27)$$

where n_s is the number of iterations performed by the subproblem approximation method and n_{si} is the number of infeasible design sets consecutively.

7.1.4 Evaluation of Design Sets

When the termination is achieved a sequence of design sets is obtained. They can be of three types:

1. Infeasible design set – it is the set, where one or more of the defined variables is not satisfied. It is out of the limited range.
2. Feasible design set – it is the set, where the all variables are in the range of their limits. All the conditions of the design are satisfied.
3. Best design set – it is the set, where the all established conditions of the design are satisfied and at the same time it reaches the lowest (minimization) or the highest (maximization) objective function value.

If the convergence is achieved during the optimization procedure, it does not necessarily mean, that the optimal solution is obtained. The optimum could be obtained only in the case that the objective function is convex and neighbouring points reach higher values of the objective function. If the objective function is not convex, the robust solution is not guaranteed. This is the reason why it is recommended to perform the optimization procedure more than once with defining of different initial feasible values. For example, define different design variable values or different range of their limits. If the design space is carefully examined and regular convergence is achieved (eqs. 7.1.22 to 7.1.25 are satisfied), from the engineering point of view the best design set is considered as the "optimum". If the optimization process is terminated from a different reason than equations (7.1.22) to (7.1.25), the best obtained design set can't be considered (if it exists) as the optimum. In that case it's necessary to perform the optimization procedure with different initial values.

7.2 FIRST ORDER METHOD

The First Order Method uses a derivation of functions to solve an optimization problem. The objective function and the penalty functions of the state variable are derived, which leads to the problem of searching a certain direction in the design space. In each iteration, a browsing of the direction by the steepest descent method and the conjugate gradient method (section 6.2.2) is performed. It means that several subiterations are performed in each iteration, which computes direction and descent of the functions.

7.2.1 Unconstrained Objective Function

The First Order Method solves all optimization problems as unconstrained optimization problems, where limit values of the design and state variables are expressed in the objective function by penalty functions. The function which solves optimization problem by the first order method has the general form

$$F(\boldsymbol{x},q) = \frac{f}{f_0} + \sum_{i=1}^{n} X_x(\boldsymbol{x}_i) + q\left(\sum_{i=1}^{m_1} W_g(g_i) + \sum_{i=1}^{m_2} W_h(h_i) + \sum_{i=1}^{m_3} W_w(w_i)\right) \qquad (7.2.1)$$

where F is the unconstrained objective function. The term X_x is the penalty function, which compensates constraints of the design variables DVs and W_g, W_h and W_w are limit values of the state variables SVs. f_0 then represents a reference objective function which was achieved in the current group of the design sets. An appropriate penalty parameter q monitors how well the design constraints are being satisfied.

Against the SAM are limiting values of the design variables expressed by the exterior penalty function method (section 6.3.2.1) and the state variables constraint by the extended interior penalty functions (section 6.3.2.3). As an example, the penalty function for the upper limit of the state variable is given:

$$W_g(g_i) = \left(\frac{g_i}{g_i + \alpha_i}\right)^{2\lambda} \qquad (7.2.2)$$

where λ is for a large positive number. Then, if the constraint is violated, the function acquires a large value. Otherwise, the penalty function acquires a very small number.

A certain advantage could be to divide the function F to two parts and separate the objective function from the penalty functions. It means, the first term is for the objective function F_f and the second is for the penalty functions, which describe the optimization problem constraints F_p. Then, the following expression is obtained:

$$F_f(\boldsymbol{x}) = \frac{f}{f_0} \qquad (7.2.3)$$

and

$$F_p(\boldsymbol{x},q) = \sum\nolimits_{i=1}^{n} X_x(\boldsymbol{x}_i) + q\left(\sum\nolimits_{i=1}^{m_1} W_g(g_i) + \sum\nolimits_{i=1}^{m_2} W_h(h_i) + \sum\nolimits_{i=1}^{m_3} W_w(w_i)\right). \qquad (7.2.4)$$

Then, the equation (7.2.1) is in the form as follows:

$$F_{(\boldsymbol{x},q)} = F_f(\boldsymbol{x}) + F_p(\boldsymbol{x},q) \qquad (7.2.5)$$

7.2.2 Direction of Searching

Every iteration (j) of the first order method computes a direction of the vector $\boldsymbol{d}^{(j)}$ for searching of the minimum of the objective function, from where the following iteration is obtained:

$$\boldsymbol{x}^{(j+1)} = \boldsymbol{x}^{(j)} + s^{(j)}\boldsymbol{d}^{(j)}. \tag{7.2.6}$$

The length of the searched line s_j from the point (which is expressed by a vector) $\boldsymbol{x}^{(j)}$ corresponds to the minimum value of F in the direction $\boldsymbol{d}^{(j)}$. The value $s^{(j)}$ is obtained by combination of a golden section search and a local quadratic fitting method (section 6.2.3.1) and it is limited by:

$$0 \leq s^{(j)} \leq \bar{s}^{(j)} \tag{7.2.7}$$

where $\bar{s}^{(j)}$ is the biggest possible step size of the search line in the current iteration j. The value $s^{(j)}$ is computed by module Design Optimization/ANSYS, but there is a possibility to affect its size by parameter S_{max}, which represents the maximum size of a step on the searched line. S_{max} is expressed in percentage. Then, the equation (7.2.7) looks like:

$$0 \leq s^{(j)} \leq \frac{S_{max}}{100}\bar{s}^{(j)}. \tag{7.2.8}$$

The solution of the minimization of the equation (7.2.1) consists in sequential generating of the line-step size $s^{(j)}$ and the appropriate penalty parameter (q). The first iteration $(j = 0)$ is solved by the steepest descent method and there is an assumption that the searched direction corresponds to a negative gradient of the unconstrained objective function.

$$\boldsymbol{d}^{(0)} = -\nabla F\left(\boldsymbol{x}^{(0)}, q\right) = \boldsymbol{d}_f^{(0)} + \boldsymbol{d}_p^{(0)} \tag{7.2.9}$$

where $q = 1$ and

$$\boldsymbol{d}_f^{(0)} = -\nabla F_f\left(\boldsymbol{x}^{(0)}\right) \quad \text{and} \quad \boldsymbol{d}_p^{(0)} = -\nabla F_p\left(\boldsymbol{x}^{(0)}\right). \tag{7.2.10}$$

The next iterations are performed by the conjugate gradient method corresponding to the Polak-Ribiere formula.

$$\boldsymbol{d}^{(j)} = -\nabla F\left(\boldsymbol{x}^{(j)}, q_k\right) + \gamma_{j-1}\boldsymbol{d}^{(j-1)} \tag{7.2.11}$$

$$\gamma_{j-1} = \frac{\left[\nabla F\left(\boldsymbol{x}^{(j)}, q\right) - \nabla F\left(\boldsymbol{x}^{(j-1)}, q\right)\right]^T \nabla F\left(\boldsymbol{x}^{(j)}, q\right)}{|\nabla F(\boldsymbol{x}^{(j-1)}, q)|^2} \tag{7.2.12}$$

If conditions of all design variables limits are satisfied, i.e. their penalty function $X_x(x_i) = 0$, the penalty parameter q could be expressed instead of the defined function F_p and can be written as follows:

$$F_p\left(\boldsymbol{x}^{(j)}, q\right) = qF_p\left(\boldsymbol{x}^{(j)}\right)\ if\ \underline{x}_i \leq x_i \leq \overline{x}_i \qquad (i = 1,2,3, \dots \dots \dots, n). \tag{7.2.13}$$

If suitable corrections are performed, the parameter q could vary its value in every iteration without any disruption of the conjugate gradient procedure, which is expressed by eq. (7.2.10). The appropriate number of the penalty parameter q allows to internally control constraints of state variables SVs. With this, if it is necessary, the constraints could be manually pushed to the state variable limit values to achieve convergence while satisfying all defined conditions. The deduction of this can be taken if the equation (7.2.10) is separated into two direction vectors. Then:

$$\boldsymbol{d}^{(j)} = \boldsymbol{d}_f^{(j)} + \boldsymbol{d}_p^{(j)} \tag{7.2.14}$$

where each of them is expressed by:

$$\boldsymbol{d}_f^{(j)} = -\nabla Q_f\left(\boldsymbol{x}^{(j)}\right) + \gamma^{(j-1)}\boldsymbol{d}_f^{(j-1)} \tag{7.2.15}$$

$$\boldsymbol{d}_p^{(j)} = -q\nabla Q_p\left(\boldsymbol{x}^{(j)}\right) + \gamma^{(j-1)}\boldsymbol{d}_p^{(j-1)}. \tag{7.2.16}$$

The whole procedure is rarely restarted with initial setting to $\gamma^{(j-1)} = 0$, due to acquisition of the steepest descent iteration. The restart is performed in one of the following cases:

1. If a discrepancy in the defined constraints is detected.
2. If the convergence of an optimization procedure is almost achieved.
3. If the satisfaction of state variable limit values is too conservative. So, if during the optimization procedure the state variable SV is far from its limit value (lower or upper), the procedure is restarted.

The first order method assumes that a direction vector exists in each performed iteration. If the vector does not exist, the optimization procedure leads to incorrect solutions. Then, the structure of the mathematical model has to be rebuilt, or a different optimization method must be used. To determine the direction vector, the following approximation is used:

$$\frac{\partial F\left(\boldsymbol{x}^{(j)}\right)}{\partial x_i} \approx \frac{F\left(\boldsymbol{x}^{(j)} + \Delta x_i \boldsymbol{e}\right) - F\left(\boldsymbol{x}^{(j)}\right)}{\Delta x_i} \tag{7.2.17}$$

where $\boldsymbol{e}$ is a vector, and all its members are equal to zero, except the ith, which is 1 and Δx_i is:

$$\Delta x_i = \frac{\Delta s}{100}\left(\overline{x}_i - \underline{x}_i\right) \tag{7.2.18}$$

where Δs is a difference between two sequential step size. In the case of using Design optimization module in the ANSYS program, Δs is defined in terms of percentage.

7.2.3 Convergence

Similarly as in SAM, the termination of the procedure comes, when the convergence criterions or conditions specified by designer are achieved. The convergence criterions are checked at the end of each performed iteration.

$$\left|f^{(j)} - f^{(j-1)}\right| \leq \tau \tag{7.2.19}$$

$$\left|f^{(j)} - f^{(b)}\right| \leq \tau \tag{7.2.20}$$

where τ is a tolerance defined in advance. The tolerance corresponds to the difference between the two objective function values. The termination also occurs if:

$$n_i = N_i \tag{7.2.21}$$

where n_i is the number of performed iterations and N_i is the number of maximum iterations defined in advance, which can be executed during the optimization process.

7.3 OPTIMIZATION TOOLS

Other optimization tools are available in the Design Optimization module, which attend to the exploration of the design space and the extreme values obtained by the optimization method. In the next section, a brief description of their algorithms and the method of searching in the design space is given.

7.3.1 Random Tool

Random Tool is a tool which has been developed to recognize behaviour and proportion of the objective function by defining of random design variable values in each performed iteration.

$$\boldsymbol{x} = \boldsymbol{x}^* = randomly\ generated\ vector \tag{7.3.1}$$

Then, if f^*, g_1^*, h_1^* and w_1^* are generated, the objective function value and the state variable values correspond to $\boldsymbol{x}^*$. Each iteration by the random tool performed presents one complete loop of the problem solution. The number of iterations is adjusted by the designer as follows:

$$n_r = N_r \tag{7.3.2}$$

$$n_f = N_f \quad if \quad N_f \geq 1 \tag{7.3.3}$$

where n_r is a number of randomly defined iterations by Random Tool performed, n_f is the total number of feasible sets (including all which were performed before Random Tool was used by any optimization tool or method), N_r is the total number of iterations by Random Tool performed and N_f is the number of feasible sets which are required.

The graphical expression of 50 random design sets performed within the frame of the case (section 9.3) is pictured in Figure 7.1.

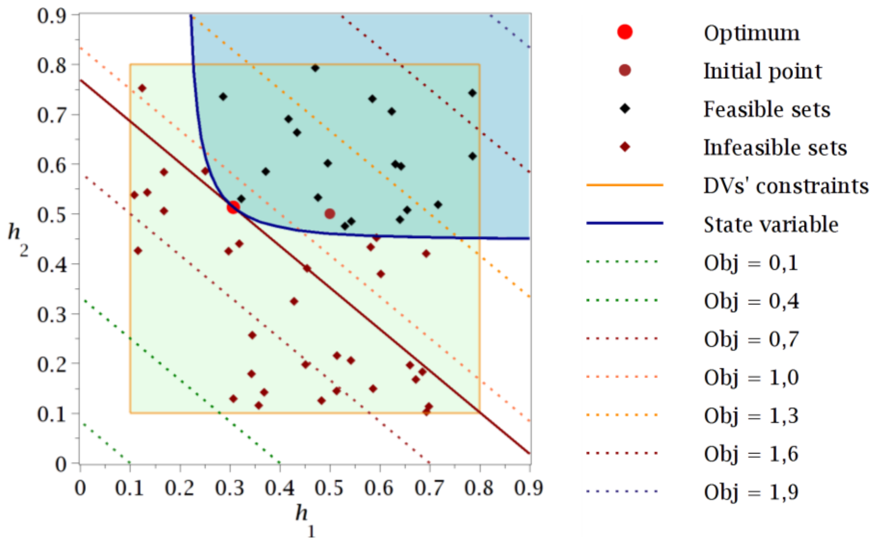

Figure 7.1 Random Tool

7.3.2 Sweep Tool

The next optimization tool is Sweep Tool. By application of the Sweep tool, the designer monitors features of the objective function in the design space by regular distribution of the design variable intervals. It means that an interval of each design variable DV is divided into sections of equal length. Then, for each section the calculation is performed with initially designed values of the other design variables. The objective function and state variable values are computed for each defined section of one design variable in the proper design variable intervals of the others (Figure 7.2). The number of iterations is determined by designer, based on an assumption, i.e. how many sections have to be performed to obtain adequate information about the objective function features in the design space. It is:

$$n_s = nN_s \qquad (7.3.4)$$

where n is a number of design variables DVs and N_s is a number of sections for each design variable, where the computation will be performed. For example, let the sweep tool design be formed for a design variable k and the final design sets established as $m+1$, $m+2$, etc., where m are all design sets, which are performed before the sweep distribution is applied. The design variables of a certain design set $m+i$ are then expressed as follows:

$$x^{m+1} = x^{(r)} + (i-1)\Delta x_i e^{(m)} \qquad (i = 1,2,3, \dots \dots \dots, N_s) \qquad (7.3.5)$$

where $x^{(r)}$ are reference design variables with x_i in j-th component of the design vector and the reference values in all the other components r are referred to the number of the reference design set. $e^{(m)}$ is a vector with 1 in m-th component and 0 in all the other components.

The increasing of the design variable value m during sweep distribution is defined as follows:

$$\Delta x_j = \frac{(\bar{x}_j - \underline{x}_j)}{(N_s - 1)} \tag{7.3.6}$$

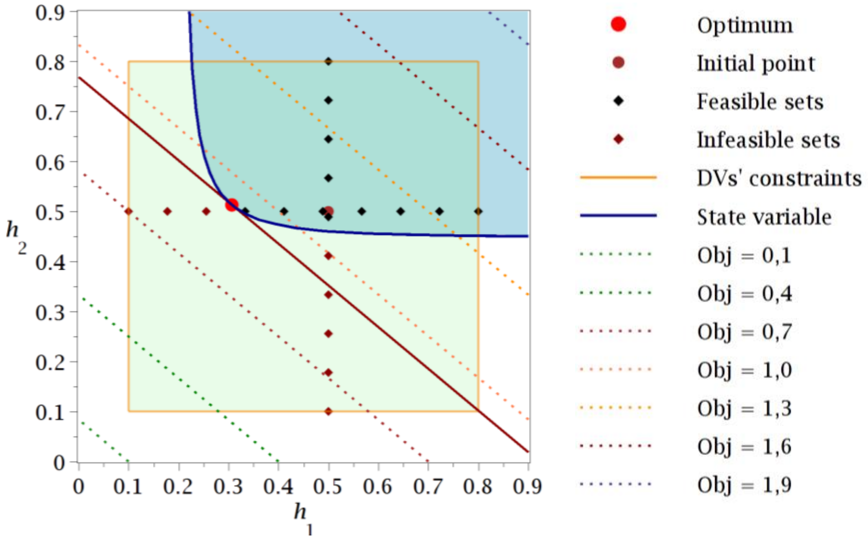

Figure 7.2 *Sweep Tool*

7.3.3 Factorial Tool

Factorial Tool is a statistical tool which has been developed for acquiring information about the progress of the optimization procedure near to the marginal points of a design space (Figure 7.3).

If the full factorial calculation is applied, with n design variables, n_{fa} of design sets are obtained, where:

$$n_{fa} = 2^n \tag{7.3.7}$$

Each design variable vector has two limit values, which are defined by interval of the certain design variable DV. That is,

$$\boldsymbol{x}_i = \bar{x}_i \text{ and } \underline{x}_i. \tag{7.3.8}$$

So, with the full factorial evaluation, all combinations of design variable limits are computed in n-dimensional design space. The solution performs the evaluation of the objective function in all the corner points of the design space.

If a fractional factorial assessment is applied, the number of generated design sets is expressed by:

$$n_{fa} = \frac{2^n}{M} \qquad (M = 2,4,8, \dots \dots \dots). \qquad (7.3.9)$$

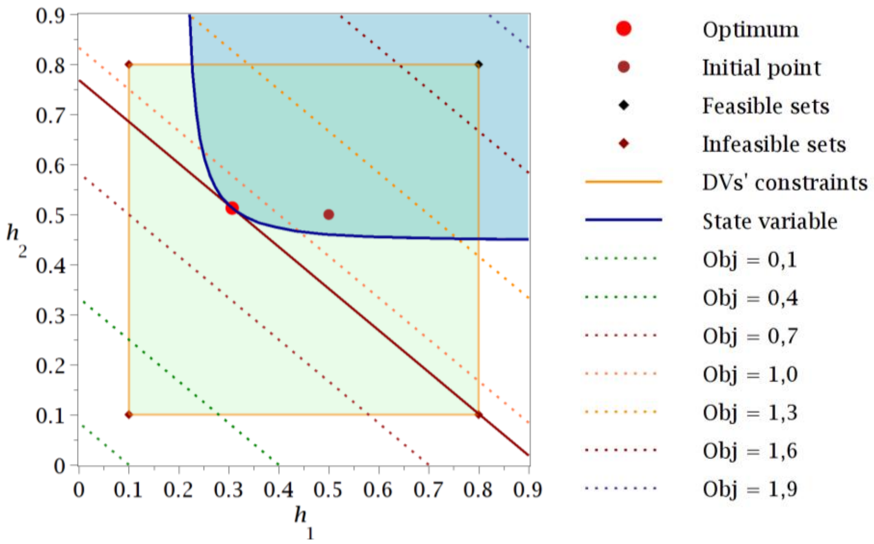

Figure 7.3 Factorial Tool

7.3.4 Gradient Tool

Gradient Tool verifies the sensitivity of dependent variables (SVs and Obj). It computes gradients of the design variables based on the defined point in the design space. The number of design sets by Gradient Tool evaluation is equal to the number of the design variables DVs, which are included in the optimization problem.

$$f_g(x) = f\left(x^{(g)}\right) \qquad (7.3.10)$$

The general expression of the objective function gradient is:

$$\nabla f_g = \left[\frac{\partial f_g}{\partial x_1}, \frac{\partial f_g}{\partial x_2}, \dots \dots \dots, \frac{\partial f_g}{\partial x_n}\right] \qquad (7.3.11)$$

Considering each design variable DV, the gradient of the objective function is expressed as follows:

$$\frac{\partial f_g}{\partial x_i} = \frac{f_g(x + \Delta x_i e) - f_g(x)}{\Delta x_i} \qquad (7.3.12)$$

where $\boldsymbol{e}$ is a vector with 1 in ith article and 0 in all the others. Δx_i is expressed by:

$$\Delta x_i = \frac{\Delta s}{100}\left(\overline{x}_i - \underline{x}_i\right) \qquad (7.3.13)$$

where Δs is the difference of the step lengths (%). The next figure (Figure 7.4) shows an application of Gradient Tool on a two-dimensional optimization problem (see section 9.3) where two design variables DVs are defined in the problem.

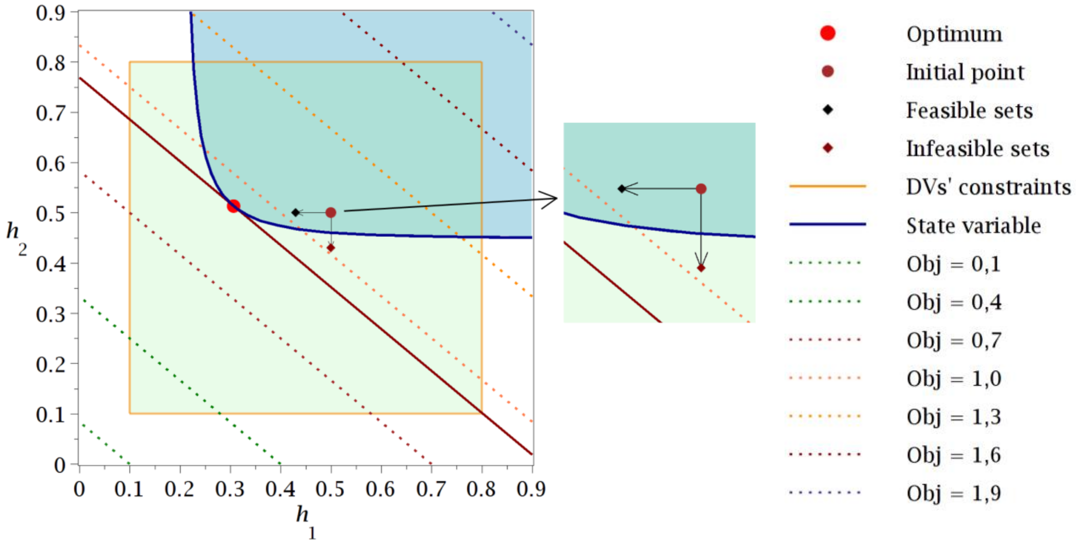

Figure 7.4 Gradient Tool

7.3.5 Single Loop Analysis Tool

Single Loop Analysis Tool is a simple and direct tool which leads a designer to understand the design space of an optimization problem. It is a suitable tool for the evaluation of the state variable SV and the objective function Obj values. Design variables DVs are always determined by a designer explicitly. One iteration by Single Loop Analysis Tool corresponds to one complete FEM analysis. The solution in the one iteration is assembled from three basic FEM parts of the computation. They are: preprocessor (model creation and boundary condition determination), solver (solving of the problem) and postprocessor (results reading module). At the beginning of each iteration, a designer defines design variable values including their limits,

$$x = x^* = design\ variables\ by\ designer\ determined \qquad (7.3.14)$$

and then one loop of the solution is performed. If the state variables and objective function are defined, the results of the solution are appropriate values of g_i^*, h_i^*, w_i^* and f* in the final design set.

8 OPTIMIZATION USING FEM/FEA

There are many accessible modern methods which can be used by designers to simulate real systems with countless amounts of technical problems. One of the most famous methods is surely the Finite Element Method/Analysis (FEM/FEA). Its broad range of utilization reaches up to many technical and even medical specializations. Among spheres where FEM analyses are actively used are the aircraft and automobile industry, mechanical engineering and to a considerable extent also civil engineering. By FEM analysis it is possible to get nearer to reality and simulate real physical (even chemical) phenomena, which are close to a real system due to a minimization of errors which usually originate from over simplifications of problem if analytical methods are used. In the design of civil engineering problems the method is used in a broad range of static and dynamic analyses, for example, stress-deflection analyses, heat transfer analyses, fluid flow analyses, acoustics, magnetic analyses, etc. Mathematically it is a numerical tool which has been developed to solve problems based on partial differential equations which describe a real system. Considering the rapid progress in computing systems and powerful computers brings more frequent usage of computing tools solving difficult technical problems. It is possible to observe progress also in an application of numerical optimization techniques in computing systems which are destined to simulate practical problems. In 1960, Lucien Schmit already foreshadowed an application of optimization techniques in structural design. He was the first one to introduce a utilization of nonlinear programming methods in a design of elastic structures. Currently many commercial programs are available based on the finite element method [31], which allow using the optimization methods for efficient design of practical problems. Specialized fields developing the efficient designs of structures using FEM are generally Design Optimization and Topological Optimization. Both of these methods use the Operating Research (chapter 4) methods to achieve an efficient design of a problem.

8.1 CLASSIFICATION OF ENGINEERING OPTIMIZATION PROBLEMS

Optimization techniques which are used in designs of technical problems can be classified according to design variables type, as *Design Optimization* and *Topological Optimization* respectively.

- The design optimization is mostly used in cases of structural problems where an objective function expresses a weight of a designed structure with the aim of its minimization. This is usually achieved by variation of a design problem shape or dimensions whose values create a design vector. The variation of design vector components must always be in accordance with all required conditions of the design. The design conditions are given by limit values of design variables and by certain restrictions of physical properties of the problem, such as stress (generally or in a given node), stiffness of a structure, local deflections, etc. The design optimization is used in other areas of civil engineering, too. For example in building physics, where the objective function with the aim of minimizing the heat flux, can be expressed by the temperature value on an external surface of a wall according to an ordination of individual layers thickness (design variables including their limit values).

- Topological optimization consists in redistribution of material. The material is centred in the most stressed sections of a structure and it becomes more compressed simultaneously with time. For that reason the other sections of the structure are less significant. The aim of the topological optimization is suitable redistribution of a material all along a surface or volume of a structure so that a weight is minimized with the possibility of most efficient usage of the material.

In the following we will put an accent only on algorithms which are used in efficient design of structures within the frame of Design Optimization.

8.2 METHODOLOGICAL PROCEDURE OF FEM/FEA

A general procedure of Finite Element Method/Analysis can be defined by the following three steps:

1. Idealization - The idealization process presents building a mathematical model which simulates important properties of a real system so that the problem in question will be sufficiently represented. Nearly always a certain amount of simplification has to be applied in model building. The simplifications must correspond to defined conditions and they cannot influence a required final value (objective function value). It means all factors which have an effect on the problem, and have minor or major influence on the final values and evaluate which simplifications can be used without violation relevancy of the real system, must be considered.
2. Discretization (meshing) - The discretization of a mathematical model consists in its division into a large number of elements which are reciprocally connected by edge or middle nodes. Each node contains a set of equations which describe its features ($ndof$ = degrees of freedom) and these features are then transferred from node to node. By this a compact set of equations arises with a finite number of elements.
3. Solution - Then, the solution consists in solving the set of equations obtained by the mathematical model discretization. The FEM/FEA procedure is schematically pictured below (Figure 8.1).

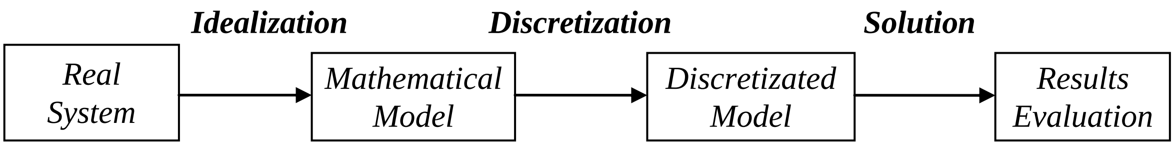

Figure 8.1 Methodological procedure scheme of FEM/FEA

A governing equation of each element in FEM/FEA is given by the following equation (8.2.1) which also represents a general static equation of FEM/FEA:

$$\boldsymbol{K} * \boldsymbol{u} = \boldsymbol{f} \tag{8.2.1}$$

where $\boldsymbol{K}$ is a stiffness matrix with dimensions $(ndof \times ndof)$, $\boldsymbol{u}$ is a displacement vector or node with $(ndof \times 1)$ displacement parameters and $\boldsymbol{f}$ is a vector of node forces with $(ndof \times 1)$ components.

In comparison a governed dynamic equation of FEM/FEA is as follows:

$$\boldsymbol{M}\ddot{\Delta} + \boldsymbol{C}\dot{\Delta} + \boldsymbol{K}\Delta = \boldsymbol{f}_t \tag{8.2.2}$$

where $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{K}$ are mass, damping and stiffness matrices, $\boldsymbol{f_t}$ is an external load vector; $\ddot{\Delta}$, $\dot{\Delta}$ and Δ are acceleration, velocity and displacement vectors of the finite element assemblage.

If dampening features are removed, then the equation (8.2.2) is the following:

$$\boldsymbol{M}\ddot{\Delta} + \boldsymbol{K}\Delta = \boldsymbol{f}_t \tag{8.2.3}$$

8.3 FEM/FEA OPTIMIZATION PROBLEM FORMULA

A general optimization problem using the finite element method can be stated as follows:

$$\mathit{Find}\, \boldsymbol{x},\ \mathit{which\ minimize}\ f(\boldsymbol{x}, \boldsymbol{u}) \tag{8.3.1}$$

subject to:

$$g_i(\boldsymbol{x}, \boldsymbol{u}) \leq 0 \qquad (i = 1,2, \dots, m) \tag{8.3.2}$$

$$h_j(\boldsymbol{x}, \boldsymbol{u}) = 0 \qquad (j = 1,2, \dots, l) \tag{8.3.3}$$

where $\boldsymbol{u}$ is the displacements vector with $(ndof \times 1)$ dimension and $\boldsymbol{x}$ is the design vector. The vector $\boldsymbol{u}$ represents implicit function of $\boldsymbol{x}$. In the case that one component x_i of the vector $\boldsymbol{x}$ changes its value, a position of the node is changed which is expressed by component u_i of the vector $\boldsymbol{u}$. The interface of $\boldsymbol{x}$ and $\boldsymbol{u}$ vectors is expressed by a partial differential equation as follows:

$$\boldsymbol{K}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{x}) \tag{8.3.4}$$

where $\boldsymbol{K}$ is a stiffness matrix $(ndof \times ndof)$, which is symmetric, band and positive-definite. $\boldsymbol{f}$ is an external force (load) vector $(ndof \times 1)$.

8.4 APPLICATION OF OPTIMIZATION ALGORITHMS IN ANSYS PROGRAM

The previous chapters denoted an access to certain optimization algorithms in efficient designs of technical problems in Design Optimization. Currently, there are many computing systems which have implemented optimization algorithms in their structures. They can be used, within the frame of the system, in efficient designs for a wide range of complicated structural problems. For this purpose the submitted work presents the ANSYS program in which an optimization module is implemented - Design Optimization module.

8.4.1 General Procedure of Design Optimization

The Design Optimization module is an individual module which is intended for solving technical optimization problems within the frame of the finite element method problems analyzed in the Ansys program. A finite element model which is subjected to an optimization procedure uses the main components of the Ansys program for model creation (model creation preprocessor), solution (solution processor) and evaluation of obtained outcomes (database results postprocessor).

If the aim of a problem is an efficient design using an optimization procedure it is necessary to consider, for the model formation, certain factors which can influence a relevancy of obtained outcomes.

Data flow during an optimization process performed in the Ansys program can be expressed by the following scheme (Figure 8.2) [75], [76]:

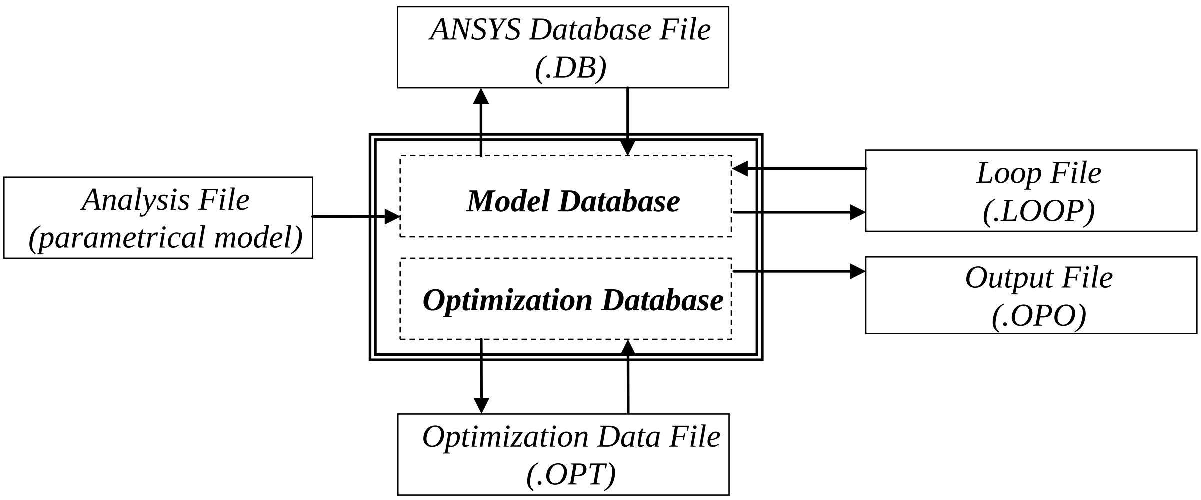

Figure 8.2 Data flow during optimization procedure in ANSYS program

The following text describes all steps which have to be done while the Design Optimization module in the Ansys program is used to achieve an efficient design of a problem.

8.4.2 Analysis File

The optimization module uses the analysis file of a problem to form an iterative procedure of an optimization process. The analysis file contains the parametrical expression of a model, parameterization of evaluated data from an initial design and an objective function. The parametrical model includes geometrical features of the model, which are used in the following as design variables - DVs (section 5.3.9.1). The evaluated data parameterization presents state variables - SVs (section 5.3.9.2). The analysis file is a key component of an optimization procedure because its content is used for creation of each iteration in the optimization process till a convergence is achieved. It must include one complete analysis from the beginning till an evaluation of outcomes and their follow-up parameterization.

8.4.2.1 Parametrical Model

The parametrical model is an expression of a problem via parameters. Each defined parameter corresponds to a quantity which varies or may vary its value during the optimization process. It means that all the defined optimization variables (DVs, SVs and objective function), and also other values which depend on them and indirectly influence the optimization procedure, are parameterized. The indirect variable can be described by an example where the distance of a column from one corner in a wall is optimized, and the wall length is fixed; then with varying of the distance the distance of the column to the second corner of the wall also changes. In this case, the first optimized distance is expressed by the design variable - DV. The second distance depends on the first one, but it does not influence the optimization procedure itself. The parametrical expression of design variables are defined by certain values before an optimization procedure starts. In comparison with it, state variables - SVs and an objective function - Obj receive their value after an initial solution is done (within the frame of the finite element method solution), thus before an optimization process starts. This means that when an optimization process starts all the variables (DVs, SVs and Obj) are already defined as parameters with certain values. The optimization variables can be defined by any quantities which influence a design, for example, geometrical parameters, material properties, boundary conditions, etc. Quantities which do not vary their values during an optimization process need not to be defined by parameters, but they must be expressed so that it is possible to identify them and create the model in each iteration of an optimization procedure. The parametrical model is created within the frame of the model creation preprocessor (PREP 7) where boundary conditions, geometrical features and material properties are defined.

8.4.2.2 Difficulties in Geometry and Discretization

Contrary to ordinarily performed analyses within the frame of FEA/FEM, there must be a special insistence on the manner of geometry creation and discretization (mesh) of a model. As mentioned before, the optimization processes are based on iterative procedures where optimization variables change their values in each performed iteration. In design optimization problems design variables often express geometrical features of a structure. Because of that the model reaches a different geometry in each iteration which certainly influences the shape or even number of finite elements which create the mesh (discretization) of the model. Hence the parametrical expression of the model must be consistent with its possible changes so that it would not, during an optimization procedure, lead to errors with sequential meaningless structure geometry or an untimely termination of the procedure. The untimely termination may occur in the case that any phase of a procedure is in discordance with logical model creation. A similar situation can occur even with defining a number and shape of the finite elements. During the optimization process some part of the geometry could be found in a situation where original definitions of mesh features are not applicable, or it reaches an unacceptable shape for correct analysis of the problem.

8.4.2.3 Solution of Parametrical Model

The following definition within the frame of the analysis file problem consists in stating a type, method and process of a solution. This means defining whether the analysis to solve the problem is linear or nonlinear, static or dynamic. Furthermore, we must also define the problem type; stability, thermal, chemical, magnetic, etc. There are also possibilities that some programs actually allow combining more different analyses. For example, frequent combinations are static and dynamic analyses, thermal and stability problems, chemical and thermal, etc. From the point of view of iterative processes it is necessary to define a manner of solution as convergence criteria, sizes of iterative steps, setup of transient process, range of frequencies in a harmonic analysis, etc. These settings are set up corresponding to the particular problem within the frame of the solution processor.

8.4.2.4 Evaluation of Parametrical Model

The last step which has to be done in the analysis file is assigning parameters to remaining optimization variables (SVs and Obj). The state variables (SVs) and the objective function (Obj) are expressed by quantities which in an optimization process represent the solution of a modelled system. These quantities and their parameterization, or rather defining optimization variables SVs and Obj, is realized after an initial solution is performed and the given quantities are obtained. The parameterization is necessary from the point of view of their reapplication during an optimization looping (iterative) procedure. An obtaining of required quantities and their parameterization is performed within the frame of the database results postprocessor (POST1).

8.4.2.5 Controlling of Analysis File

In this phase of the process the verification of the relevancy of defined parametrical expression to the given problem is very important. The recommendation is to perform one independent computation of the defined parametrical model and test correctness by manual calculation or through other available methods. That means all the steps of model creation described above represent one compact analysis, including parameterization of all quantities, which directly or indirectly play a role in the optimization process. If the verification is not performed or it is not substantial the incorrect results could be obtained or a convergence of the optimization procedure would not be achieved. The Design Optimization module must have clear access to the parametrical model and its solution so that all needed iterations can be performed to achieve required outcomes.

8.4.3 Design Optimization Module Initiation

The next step of the process is already included in the Design Optimization module. In the first place there has to be clear access to the (in advance created) analysis file which includes a definition of the parametrical expression of the complete FEM/FEA analysis. Then the optimizer immediately presents defined parameters as input values of the optimization process. This means that it creates the first optimization design set (section 5.3.3) which represents the start point for the optimization procedure.

8.4.4 Defining of Optimization Variables

So far the model is formulated by parameters with which the Design Optimization module is not able to work. It is necessary to assign the parameters to the certain optimization variables (design variables DVs, state variables SVs and objective function Obj). Then these are finally used for a mathematical model formation of a defined problem which corresponds to the general optimization problem form (eqs. 4.3.1). Then the mathematical model is subjected to the optimization procedure. The optimization variables DVs and SVs represent the defined parameters which are defined in the analysis file. They control the entire optimization process and influence a final objective function value. The Design Optimization module in the Ansys program allows defining up to 60 design variables DVs, 100 state variables SVs and one objective function Obj. The optimization algorithms which are implemented in the Design Optimization module are not intended to solve multi-objective optimization problems following prescribed Design Optimization modules' instructions.

Nevertheless, in some cases if certain conditions are considered and proper variables with their limitations are defined, these problems also can be solved using the presented optimization module.

8.4.5 Specification of Optimization Method or Tool

The technique of exploration of a design space and manner of searching an optimal solution depends on the choice of an optimization tool or method. The Ansys/Design Optimization module allows solving minimization or maximization problems with two optimization methods and four optimization tools. The chapter 7 deals with their detailed description. Although the optimization tools are applied to the initial exploration of a design space (Random Tool, Sweep Tool, Factorial Tool) or testing and proving of achieved outcomes (Gradient Tool) they can be occasionally used to find an extreme point of an objective function (Obj). If any tool is not adequate to solve the problem the optimization method (First Order Method, or Subproblem Approximation Method) is used. Thus the designer must select a suitable method or tool to solve the given optimization problem. If the problem is not very time-consuming it is possible to try more or even all the available techniques and then choose the best achieved design set. But in most design problems, time plays a significant role and a designer needs to choose a suitable tool or method to achieve an efficient and less time-consuming solution of the optimization problem. Features of the optimization methods and tools which are implemented in the Ansys/Design Optimization module are explored in chapter 9, A Set of Case Problems and Their Numerical Solutions. In the case that the objective function features and its approximate extreme, both are known, it seems to be suitable to use the First Order Method (FOM) which generally reaches more accurate results, although to the detriment of the solutions' time-consumption. Otherwise, if the objective function behaviour is not known or it is known that the objective function reaches more local extremes in a certain interval, it is recommended to use the Subproblem Approximation Method (SAM). If the SAM method is used it is advisable to explore a problems' design space with one of the available optimization tools. Advantages and disadvantages of the methods and tools are described in chapter 7, Design Optimization Methods and Tools.

8.4.6 Specification of Method or Tool Settings

Each of the optimization methods and tools requires specific access in defining parameters which influence their approach to a problem. In all of them, except Sweep Tool and Factorial Tool, it is possible to define minimal or maximal number of iterations (loops), or minimal or maximal number of feasible or infeasible design sets. The number of iterations within the frame of Sweep Tool and Factorial Tool depend on the number of design and state variables (section 5.3.9). If the FOM (First Order Method) method is used, step length in a direction of procedure in a design space can be influenced by a designer in each iteration. On the other hand, the SAM (Subproblem Approximation Method) method allows defining the importance of certain optimization variables via functional weights. These methods (FOM and SAM) are described in detail in chapter 7. A convergence criteria ε are defined the same for all the available methods and tools and also a manner of results' design sets obtaining. A brief summary of specifications using all the methods and tools is pictured bellow (see Table 8.1 and 8.2) [75].

Table 8.1 Specification of Methods' settings

Optimization method	Setting specifications
First Order Method	*-maximal number of performed iterations* *-limit which is applied to the size of each line search step. It defines a limit range of design variables changes for each iteration.* *-shift of a design variable range which is used to compute a gradient Δ in the form:* $\frac{\Delta\left(g_i^{max} - g_i^{min}\right)}{100}$

Subproblem Approximation Method	*-maximal number of performed iterations*
	-maximal number of infeasible design sets within the optimization procedure
	-controlling of curve fitting of an approximated function
	  -fitting of an objective function Obj (linear, quadratic)
	  -fitting of a state variable SV function (linear, quadratic)
	-weighting factors applied to
	  -distances of design sets in a design space
	  -objective function values
	  -feasibility or infeasibility of design sets
	-approximation reformulation can be executed
	  -in each iteration
	  -each nth iteration

Table 8.2 Specifications of Tools' settings

Optimization tool	Setting specifications
Sweep Tool	*-reference point specification* *-number of iterations performed for each design variable in sweep distribution over its range*
Factorial Tool	*-type of factorial evaluation* $\left(1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}\right)$
Gradient Tool	*-defining of reference point for gradient evaluation (number or specification of a design set)* *-shift of a design variable range which is used to compute a gradient Δ in the form:* $\frac{\Delta\left(g_i^{max} - g_i^{min}\right)}{100}$
Random Tool	*-maximal number of executed iterations* *-minimal number of feasible design sets (including all which have been achieved before)*

8.4.7 Initialization and Termination of Optimization Procedure

An optimization procedure can be initiated, when all the previous steps are successfully performed. The optimization procedure if one of the optimization methods is used, consists of an iterative evaluation of a minimum or maximum value of an objective function (section 5.2). In the case that an optimization tool is applied, the evaluation consists of an iterative process by prescribed procedures (section 7.3). If previous steps do not signify any problem the optimization process is terminated by convergence criteria (sections 7.1.3 and 7.2.3), or after the specified number of iterations is achieved. Nevertheless, although faultless formulation of the problem exists, a designer might meet error messages which alert about some discrepancies in the model creation. Mostly the discrepancies are due to the incorrect generating of a mesh (finite elements). It could be inflicted by a geometry change of the problem which changes its shape in each iteration (if the optimization depends on the geometry). With the structures' geometry change, there is also a change in shape or number of the finite elements (depending on their definition) within the frame of the analyzed problem. If the optimization process is terminated because of mesh errors, it is necessary to reformulate the finite elements generation and repeat the entire process again. Another unexpected cause of preliminary termination might be a divergence of the problem, for example during nonlinear problem solution. If the defined number of iterations is achieved or the evaluation is terminated by satisfying convergence criteria, the designer approaches the next phase of the process. The next phase depends on the method or the tool which has been used for the present evaluation. If the goal of the previous analysis has been to explore a design space by an optimization tool the consequential process consists in evaluation of the objective function minimization (maximization) using an optimization method (SAM or FOM), considering outcomes which have been obtained in previous analysis. When the optimization process has been performed using an optimization method and the required number of iterations or convergence criteria has been achieved, the designer decides if the following procedure continuous by an evaluation of obtained design sets or the obtained minimal (maximal) point's vicinity should be explored using Gradient Tool (section 7.3.4).

8.4.8 Evaluation of Design Sets Data

After the optimization process is completed the designer must verify the accuracy of the obtained outcomes and evaluate if the required results have been achieved. The result of an optimization procedure is a series of design sets. Two types of design sets can be obtained, feasible and infeasible (section 5.3.13). The feasible design set which achieves the lowest (minimization) or the highest (maximization) objective function value is considered the optimum. This means that it satisfies all defined conditions of optimization design variables and simultaneously reaches the lowest or the highest value from all the obtained design sets. In the case that a suitable solution has not been obtained the designer has to evaluate the situation and decide how to proceed. Problems can occur in some contradictions within the frame of an initial design, in inappropriately defined limit values of optimization variables, or any other problem which can terminate the optimization procedure for a different reason than those which were defined by the designer. In the case that the required design sets were obtained and the convergence criteria were satisfied, the designer searches the design set which achieves the best satisfaction of all the conditions (feasible design set). Then the design set is considered as the result (optimum) of the solution. It must be noted that in engineering problems it is almost impossible to obtain the optimal solution (with regard to manufacturing procedures and to very difficult consideration of all factors which influence the real system), but the usual aim is to achieve a result which is as near to the optimum as possible. It is recommended to repeat the optimization process considering a different initial point than the actual, or if needed using a different optimization method or tool because of possible bogging in a local extreme. If detailed analysis has been performed, all the conditions and criteria are satisfied and the designer is convinced that the required extreme of the objective function has been achieved, then the parameter values of the best design set will replace suitable values in the initial problem definition, which is then considered as the efficient design of the problem given by the defined conditions.

9 A SET OF CASE PROBLEMS AND THEIR NUMERICAL SOLUTIONS

In the following, optimization methods which are included in an optimization module Design Optimization/ANSYS are analyzed. The aim of the analyses is the verification of presented optimization techniques and their applicability in designing of practical engineering problems. The stress was put especially on relevancy and robustness of the methods.

The presented optimization techniques are First Order Method (FOM) and Subproblem Approximation Method (SAM). Each of these methods consists of different techniques of searching an extreme of an optimization problem. The FOM method is based on derivative approaching of the extreme with the aid of the steepest descent method and the conjugate gradient method (section 6.2.2) and the SAM method consists of searching for the extreme by sequential unconstrained minimization technique (section 6.3.3) of dependent variables' approximated functions performed by least squares fitting (section 6.2.1.1).

Both methods require specific definitions of the optimization problems to achieve efficient progress in structural designing. The robustness and efficiency of the methods are controlled by optimization tools (Sweep, Random, Gradient, Factorial - section 7.3).

The formulations of the optimization problems are chosen so that the methods' proceedings are controllable by manually computed and/or graphical solutions.

The application of the methods in efficient design of a truss-beam and air gap location in wooden studs which represent multi-variable optimization problems is presented at the end of this chapter.

9.1 GRADIENT AND APPROXIMATION METHODS' ROBUSTNESS - LOCAL VS. GLOBAL EXTREMES

Many engineering optimization problems are characterized by strict convex or concave functions. These lead a designer to find one and only one extreme of an objective function. On the other hand, problems which cannot be or are very difficult to prove by graphical or mathematical evaluation existence of the one extreme are suitable to explore design space and search for other possible extremes. Multi-extreme optimization problems must be considered with caution, responsibility and analysis deep enough to avoid achieving local extremes of a final function (robust design). Hence the robustness of presented methods is analyzed through the following multi-extreme optimization problem where the objective function is defined by two global and four local minimums within the frame of specified design space. The objective function is graphically expressed in Figure 9.1. The problem is solved in advance by combination of computational and graphical solutions whose results are then used for verification of design sets' values from optimization process achieved by the Design Optimization module. The Subproblem Approximation Method (SAM), First Order Method (FOM) and optimization tools (Random, Sweep, Factorial) are analyzed within the frame of the presented problem.

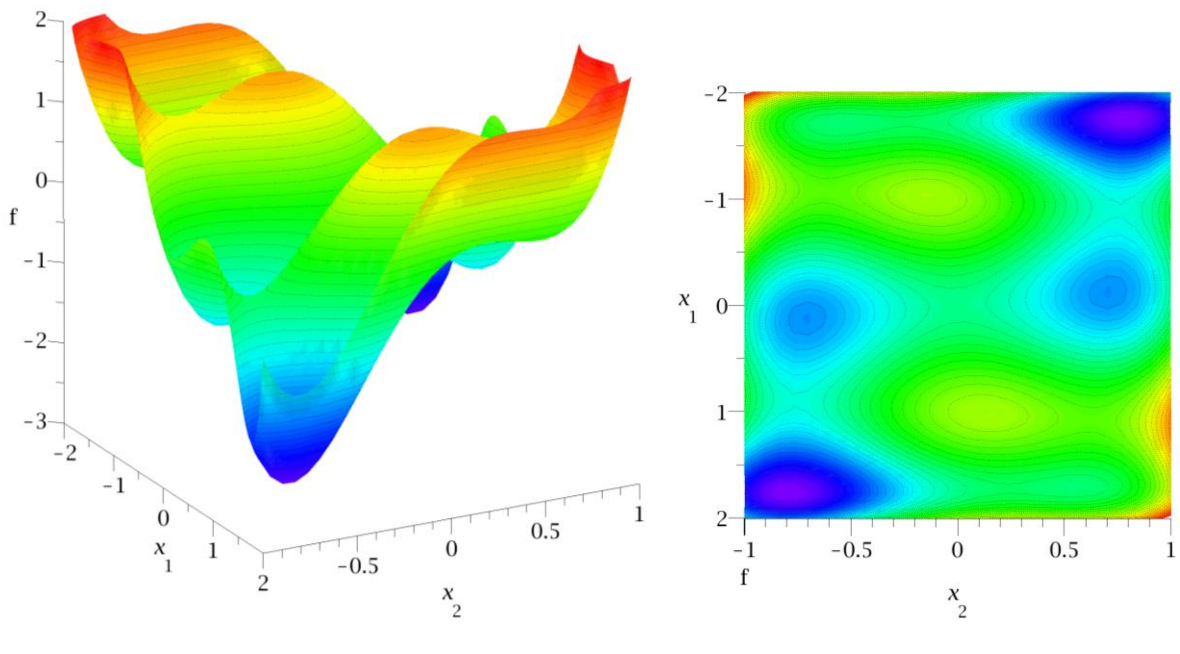

Figure 9.1 Graphical expression

9.1.1 Problem Definition

The aim of the problem is to find minimum of the following function:

$$f(x,y) = -2x_1^4 + 4x_2^4 + \frac{x_1^6}{3} + 3x_1^2 - 4x_2^2 + x_1x_2. \qquad (9.1.1)$$

Limiting conditions of the optimization problem are defined by lower and upper values of design variables DVs as follows:

$$\begin{aligned} -2 \le x_1 \le 2 \\ -1 \le x_2 \le 1 \end{aligned}. \qquad (9.1.2)$$

Then the optimization problem is defined:

$$Find\,\boldsymbol{x} = \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix},\ which\ minimize \quad f(\boldsymbol{x}) = -2x_1^4 + 4x_2^4 + \frac{x_1^6}{3} + 3x_1^2 - 4x_2^2 + x_1x_2 \qquad (9.1.3)$$

subject to:

$$\begin{aligned} -2 \le x_1 \le 2 \\ -1 \le x_2 \le 1 \end{aligned}. \qquad (9.1.4)$$

9.1.2 Localization of Extremes

The problem is expressed in the defined design space by four local and two global minimums. According to graphical expression (Figure 9.1) and bisection method the values of all minimums are evaluated. They are:

Table 9.1 Global and local minimums

Variable	Global minimum		Local minimum 1		Local minimum 2	
f (Obj)	-2,3199		-1,0425		0,1020	
	1st	**2nd**	**1st**	**2nd**	**1st**	**2nd**
x_1 (DV)	-1,7628	1,7628	-0,1215	0,1215	1,7074	-1,7074
x_2 (DV)	0,7987	-0,7987	0,7146	-0,7146	0,5546	-0,5546

9.1.3 Solution by ANSYS/Design Optimization

The problem is subjected to analyses performed by the FOM and SAM methods. The effectiveness of the methods in solving multi-extreme problems is tested. Both methods are controlled within the frame of different settings of internal parameters which influence the techniques' proceedings.

The Design Optimization module requires positive design variables' values (*DVs>0*). For this reason the optimization problem (eqs. 9.1.3 and 9.1.4) have to be redefined. The optimization problem is then defined as follows:

$$Find\,\boldsymbol{x} = \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix},\ which\ minimize\quad f(\boldsymbol{x}) = -2x_1^4 + 4x_2^4 + \frac{x_1^6}{3} + 3x_1^2 - 4x_2^2 + x_1 x_2 \qquad (9.1.5)$$

subject to:

$$\begin{aligned} -2 &\le x_1 \le 2 \\ -1 &\le x_2 \le 1 \end{aligned}. \qquad (9.1.6)$$

Within the frame of the Design Optimization module the DVs' conditions of the optimization problem are formulated:

$$\begin{aligned} 1 &\le h_1 \le 5 \\ 1 &\le h_2 \le 3 \end{aligned} \qquad (9.1.7)$$

where

$$\begin{aligned} x_1 &= h_1 - 3 \\ x_2 &= h_2 - 2 \end{aligned} \qquad (9.1.8)$$

In solutions performed by the optimization tools (Random, Factorial, Sweep), FOM and SAM methods the design variables are expressed by h_1 and h_2.

9.1.4 Factorial Tool

The Factorial Tool allows the designer to recognize behaviours of dependent variables in marginal points of the specified design space (section 7.3.3). This tool is especially effective for suitable election of limit values of the optimization problem. In the presented problem this is not required because the design space is determined by design variables' upper and lower limits only and state variables are not defined. This means that the design sets according to marginal points of the design space represent a feasible solution.

9.1.5 Sweep Tool

Sweep tool leads a designer to explore the objective function by varying one design variable (section 7.3.2). The range of each design variable is divided into the same number of sections depending on the designer's choice. The Sweep Tool allows definition up to 9 sections for each range of defined design variable which corresponds to 10 performed loops for each design variable. The presented constrained optimization problem was subjected to the Sweep Tool considering the initial point $\boldsymbol{x} = \{-1; -1\}^T$ (Figure 9.2).

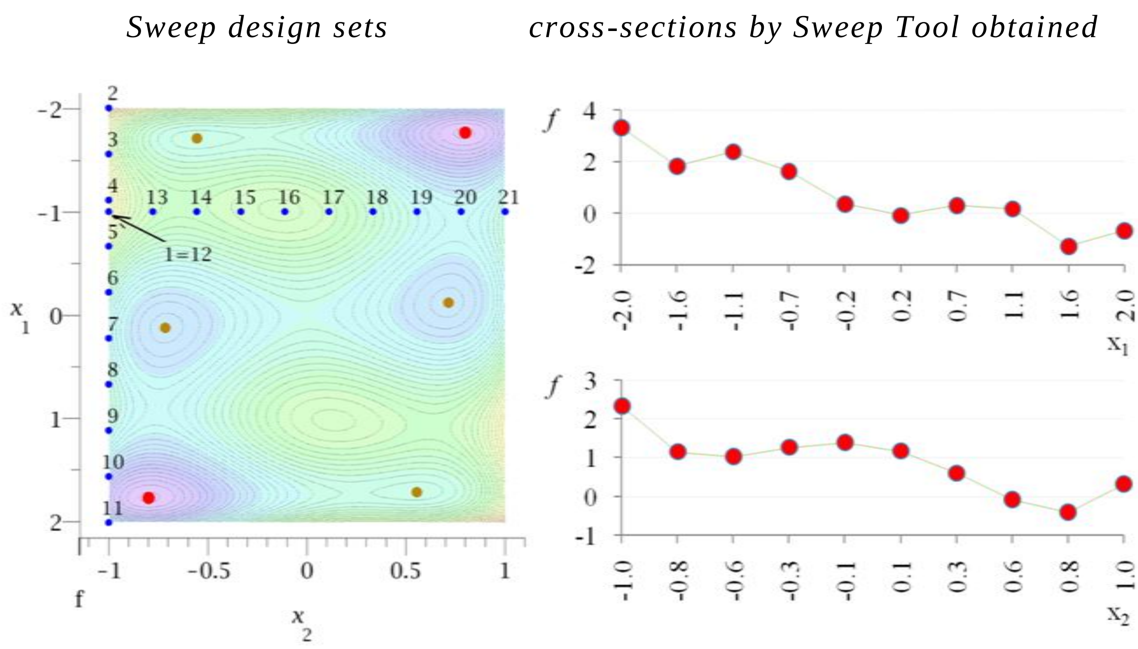

Figure 9.2 Sweep Tool - Design variables' location

The controlled exploring of a design space by Sweep Tool is suitable especially in the case where an extreme existence in a certain location is suspected. Then the Sweep Tool allows uniform distribution of design sets within the frame of the design directed by a defined initial point.

9.1.6 Random Tool

Random Tool defines random design sets within the frame of the design space (section 7.3.1). The tool applied in the presented problem shows satisfactory coverage of the entire defined design space. It follows that the tool can be effectively used for exploring the design space before the optimization methods are applied. With increase of random loops performed the chance to approach ambient of the global extreme also increases. Distribution of 100 randomly defined design sets within the frame of presented problem is pictured in Figure 9.3. The random design sets give good initial points to the SAM method to approximate dependent variables with widely distributed points, and also, if the best design set by Random Tool obtained is in ambient of the global extreme, the FOM method can be effectively applied to achieve its accurate location.

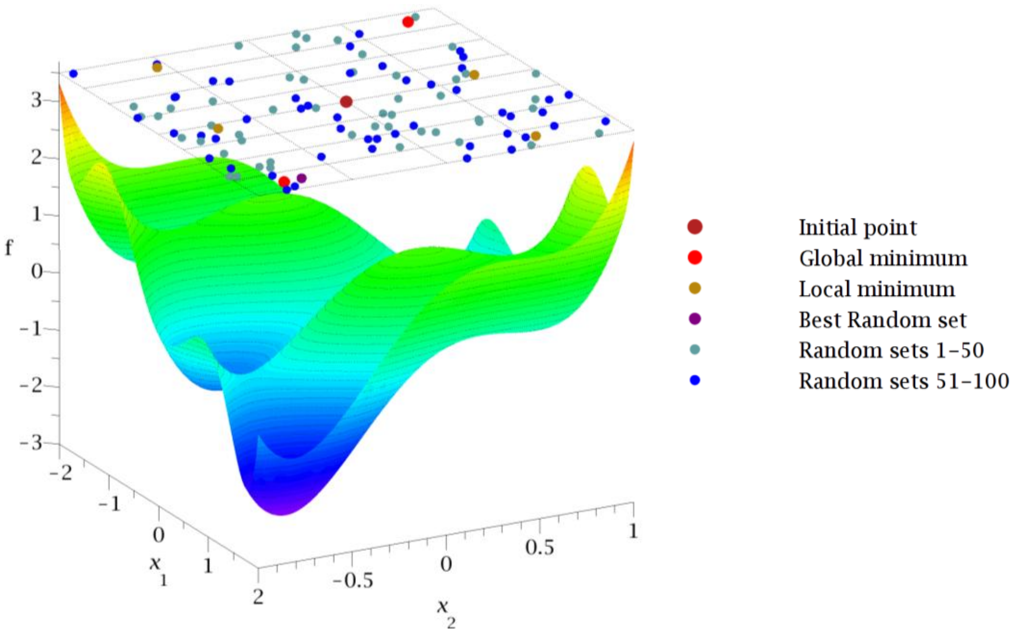

Figure 9.3 Random Tool - 100 randomly performed loops

9.1.7 First Order Method

In the following, the robustness of the FOM method is analyzed depending on different initial points' location within the frame of the design space without its being explored by an optimization tool. The optimization process by FOM was applied for each initial point pictured in Figure 9.4 with varying step lengths' range of gradients. Their maximum step length (section 7.2.2) is divided into 10 even sizes, which means 450 solutions by the FOM methods were performed.

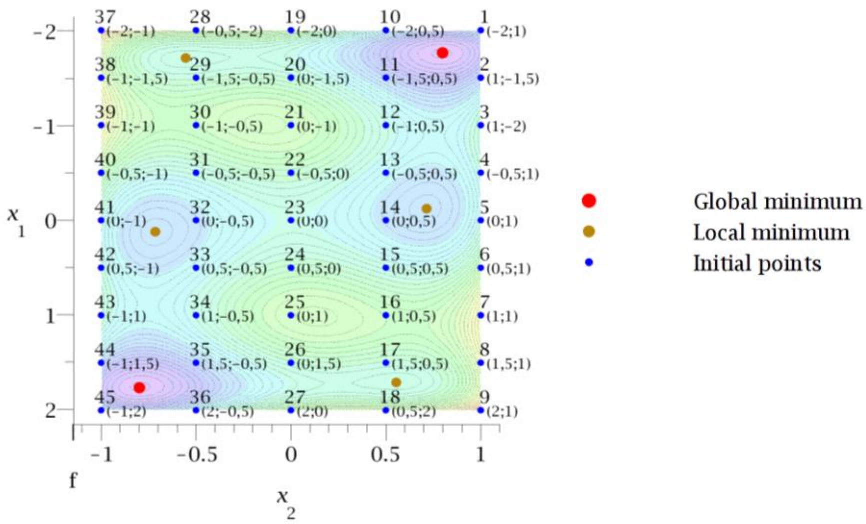

Figure 9.4 Initial points which are used for analyses

The following table (Table 9.2) shows variables' values in the best design set obtained in each solution for all specified initial points defined in Figure 9.4.

Table 9.2 FOM analysis

Initial point	**Step length [%]**	x_1	x_2	**Best set number**	**Total number of sets**	***f* value in best set obtained**	**Convergence status**
1	90	-1,7685	1,0000	2	5	-1,7515	↑
2	10-100	-1,5000	1,0000	1	3	-1,0781	↑
3	90	-1,7643	0,7996	6	9	-2,3199	↓
4	30	-0,1234	0,7143	4	7	-1,0425	→
5	100	-0,1217	0,7140	3	5	-1,0425	→
6	30	-0,1278	0,7153	4	6	-1,0424	→
7	100	0,1239	-0,7144	4	6	-1,0425	→
8	80	0,1227	-0,7184	7	10	-1,0424	→
9	100	-0,1216	0,7144	4	6	-1,0425	→
10	10-100	-2,0000	0,5000	1	3	-0,4167	↑
11	10-100	-1,5000	0,5000	1	3	-1,0781	↑
12	100	-1,5605	1,0000	3	76	-1,3015	→
13	10-100	-0,5000	0,5000	1	3	-0,3698	↑
14	10-100	0,0000	0,5000	1	3	-0,7500	↑
15	80	-1,7636	0,7995	3	5	-2,3199	↓
16	50	-0,1321	0,7143	4	7	-1,0422	→
17	90	-1,6250	0,9344	4	130	-1,8480	→
18	90	-1,7611	0,8106	5	7	-2,3182	↓
19	50	-1,7339	0,5017	3	5	-1,6233	→
20	90	-1,7652	0,8006	6	8	-2,3198	↓
21	20	-1,7628	0,7899	13	16	-2,3190	↓
22	90	-0,1011	0,7235	4	6	-1,0405	→
23	10-100	0,0000	0,0000	1	3	0,0000	↑
24	90	0,1046	-0,7218	4	6	-1,0412	→
25	10	1,7639	-0,7957	24	26	-2,3198	↓
26	100	1,7611	-0,8010	6	8	-2,3198	↓
27	40	1,7373	-0,4060	3	5	-1,2556	→
28	90	1,7633	-0,8047	5	7	-2,3195	↓
29	80	0,1232	-0,7152	5	7	-1,0425	→
30	100	0,1191	-0,7175	5	7	-1,0424	→
31	90	1,7631	-0,7983	4	7	-2,3199	↓
32	10-100	0,0000	-0,5000	1	3	-0,3698	↑
33	10-100	0,5000	-0,5000	1	3	-0,3698	↑
34	100	1,5447	-0,9980	3	51	-1,2578	→
35	10-100	1,5000	-0,5000	1	3	-1,0781	↑
36	10-100	2,0000	-0,5000	1	3	-0,4167	↑
37	100	1,5820	-0,9980	3	5	-1,3885	→
38	70	-1,7641	0,8210	4	6	-2,3141	↓
39	100	-1,7578	0,7665	3	6	-2,3084	↓
40	70	-0,1215	0,7123	5	7	-1,0425	→
41	90	0,1209	-0,7134	4	7	-1,0425	→
42	30	0,1214	-0,7144	4	7	-1,0425	→
43	100	1,7630	-0,7989	5	8	-2,3199	↓
44	10	1,4403	-0,8419	2	5	-1,4459	↑
45	50	2,0000	-0,7639	2	8	-1,1666	↑

→ convergence to local minimum ↓ convergence to global minimum ↑ divergence

Although the problem seems to be symmetrical and the initial points 1 to 22 reflect the points 24 to 45, subtle distinctions were achieved in obtained results.

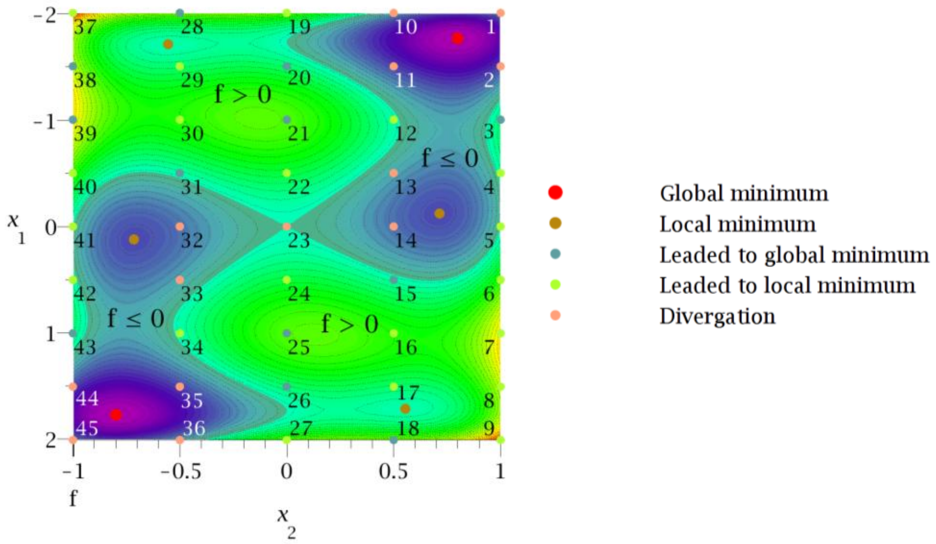

Figure 9.5 FOM analysis with different initial points' location

The performed analysis of the FOM method shows it is not suitable to define initial points in the location where the objective function achieves negative values although the design variables are determined as positive. If the original formulation of the objective function is defined so that it is specified by negative values locally or globally, it is recommended to redefine it so that the objective function reaches positive values all over the design space. The solutions where the initial point corresponds to the negative objective function diverged (Figure 9.5). However, the remaining initial points led to a certain progress of finding the extreme of the problem. The progresses which were individually obtained by the solutions for each initial point are pictured in the following table (Table 9.3):

Table 9.3 Convergence of solutions

Obtained solution	Number of initial point (Figure 9.4)
Global minimum	3; 15; 18; 20; 21; 25; 26; 28; 31; 38; 39; 43
Local minimum	4; 6; 7; 8; 9; 12; 16; 17; 19; 22; 24; 27; 29; 30; 34; 37; 40; 42
Divergence	1; 2; 5; 10; 11; 13; 14; 23; 32; 33; 35; 36; 41; 44; 45

The following figures (Figure 9.6) show progress of the FOM method in the initial points where one of the global minimum was achieved.

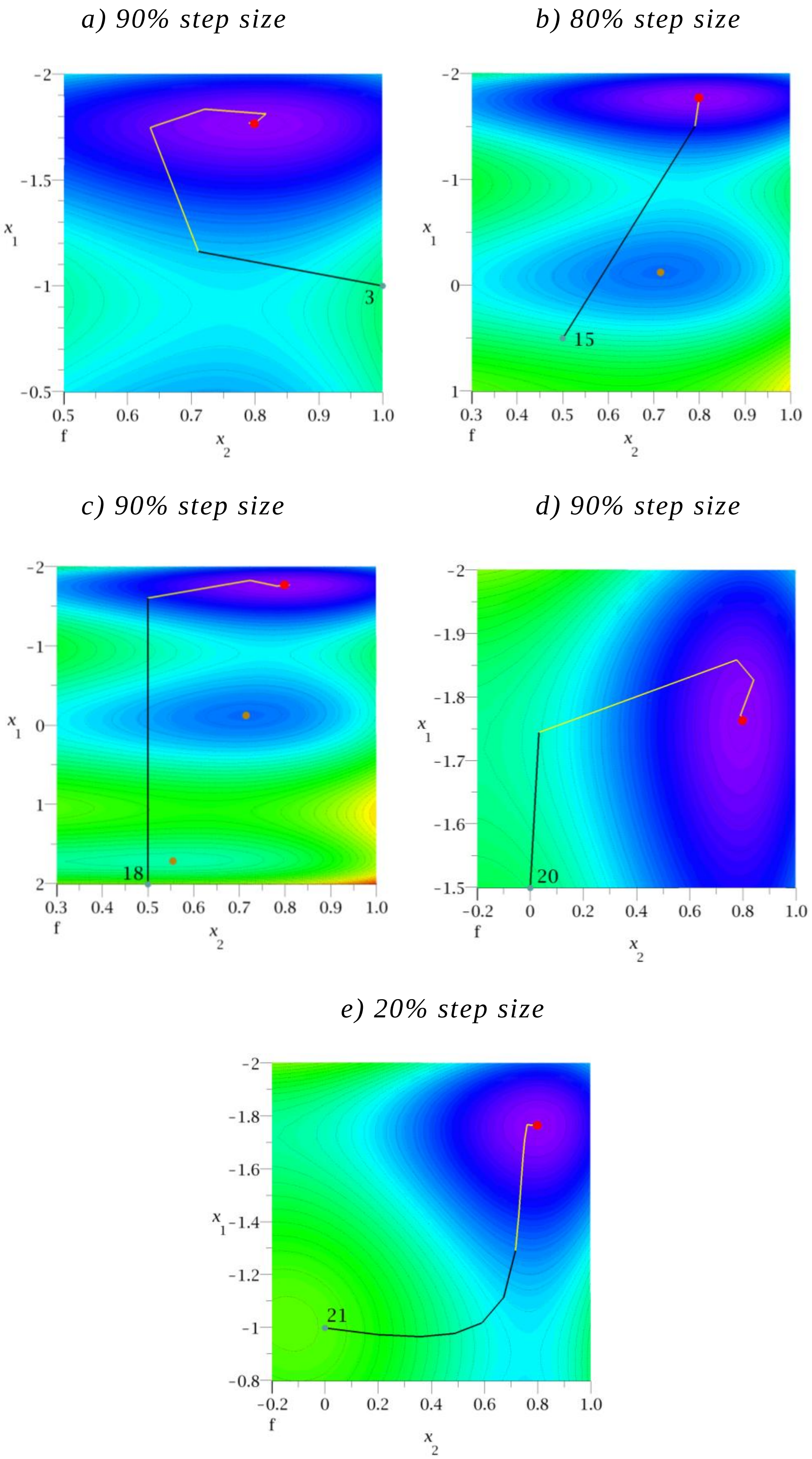

Figure 9.6 FOM method progress in different initial points

The robustness of own FOM method depends on choice of the initial point, its character of the objective function in ambient, and internal properties' definition of the method. If the initial point is chosen near to a local extreme of the problem and short length of step size is defined the solution tends to the local extreme. In the same case but with definition of longer step size the solution might converge to a different location in the design space than to the nearest extreme. If the character of the objective function is unknown and using only the FOM method is required the robustness of the solution is not guaranteed because a local extreme can be found. From this point of view it is suitable to explore the design space by one of the optimization tools (section 7.3).

9.1.8 First Order Method Applied in Explored Design Space

In the following the FOM method is analyzed with exploring of the design space. The FOM algorithm initiates in the best achieved design set which occurs in the actual optimization database. The previous analysis shows (sections 9.1.4 - 9.1.6) that the most suitable tool for maximizing robustness of the optimization performed by FOM would be the Random Tool. By using the Random Tool there is quite high possibility that one or more design sets are located near to the global extreme. One of the most important parameters which must be observed within the frame of the FOM method is step length. The default step length setting corresponds to maximal length (section 7.2.2) computed by combination of golden section search and local quadratic fitting method (section 6.2.3.1). The length can be influenced by a designer in range $\langle 0; 100\rangle$% from the maximal length. In the presented problem the 10, 20, ...100 % of the maximal step length were analyzed. The picture (Figure 9.7) shows the progress of the FOM method where the best solution by the Random Tool (section 9.1.6) obtained is defined as initial design set for 10 and 100% of maximal step length. In this case the initial point is already very close to the optimum. This causes the solution where long step length is applied to jump over the global minimum and tends towards a different location in the design space. In this case it converges in a local minimum (Figure 9.7). With application of short step length the solution points to the global minimum which is in ambient of the initial point. On the other hand, if the initial point is near to a local minimum short step length causes finding the local minimum, whereas long step length solution might find the global minimum of the problem. The efficiency of the FOM method consists of exploring the design space by any method which allows approximately uniform diffusion of design sets in the design space. Then the best achieved design set in the explored design space might be a suitable initial point for the FOM method proceeding with short step length definition to search the global extreme. The robustness of the solution depends on a number of explored design sets and their layout in the design space. With increasing of the explored design sets a probability that one of the point is near the global minimum increases simultaneously.

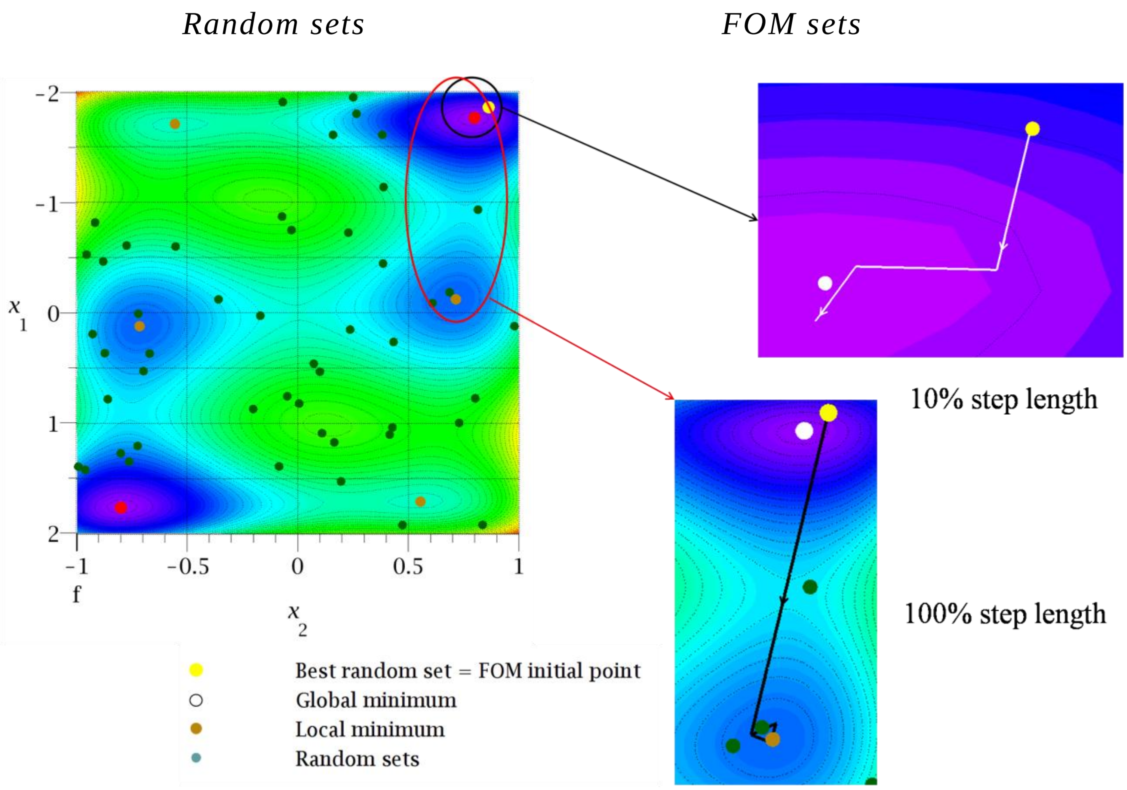

Figure 9.7 FOM with random exploration of design space

9.1.9 Subproblem Approximation Method

In the case where the SAM method is applied the definition of an initial point doesn't influence the optimization proceeding as considerably as in the case of the FOM method. The effect of the initial point selection depends on a number of loops which are performed before actual iteration by the SAM method is initiated. With increase of the design sets' number the effect of the initial point location decreases. This is inflicted by forming of the objective function approximation which is based on design sets achieved beforehand. The optimization loops (design space exploration) can be performed by one of the optimization tools (see section 7.3). Each of the optimization tools allows defining a certain number of loops to perform. If the maximal number of loops is not satisfying a combination of two or more tools can be determined. If any of the optimization tools is not chosen four loops by the Random Tool are performed by default before the SAM method initiates. In the following the SAM method is analyzed depending on different features of dependent variables' (Obj and SVs) approximation fitting and weighting factors. The achieved results are summarized in the tables below (Tables 9.5 - 9.6). The SAM method processing features are marked by superscripts where the superscripts' specifications are pictured in the following table (Table 9.4).

Table 9.4 SAM analysis

Setting	Superscript	Conditions
Obj (Objective function fitting)	0	Quadratic + cross-term curve
	1	Linear curve
	2	Quadratic curve
SVs (State variables fitting)	0	Quadratic curve
	1	Linear curve
	3	Quadratic + cross-term curve
W (Weighting factors)	0	Design space, Obj and feasibility of solution
	1	All are unity
	2	Distance in design space
	3	Obj (Objective function)
	4	Feasibility/infeasibility of solution

At first the SAM method is initiated by 5 random loops. A combination of all different setting possibilities leads to performing 45 computations. On the other hand the solutions can be distributed into 11 groups according to defined optimization problem and achieved results. The presented problem is defined by only one dependent variable (Obj) which means that approximation method of state variable doesn't influence optimization process in any way. Then three categories can be expressed according to objective function approximation method. The results from the SAM method solution initiated by 5 random design sets are pictured in the following table (Table 9.5) and locations of obtained design sets are graphically expressed in Figure 9.8.

Table 9.5 SAM analysis

Category	Group	Settings	f	x_1	x_2	Iterations
Initial point			*0,0000*	*0,0000*	*0,0000*	
1	1	$(Obj^{0},SVs^{0,1,3},W^{0})$	-2,3199	1,7627	-0,7987	261 (1982)
	2	$(Obj^{0},SVs^{0,1,3},W^{1,4})$	-1,6563	1,6707	-0,9958	9 (2622)
	3	$(Obj^{0},SVs^{0,1,3},W^{2})$	-1,7759	1,7638	-0,9960	135 (2474)
	4	$(Obj^{0},SVs^{0,1,3},W^{3})$	-2,3199	1,7628	-0,7987	275 (1900)
2	5	$(Obj^{1},SVs^{0,1,3},W^{0,3})$	-1,7336	1,8243	-0,9953	7 (2222)
	6	$(Obj^{1},SVs^{0,1,3},W^{1,4})$	-0,7039	1,3914	-0,9933	5 (3430)
	7	$(Obj^{1},SVs^{0,1,3},W^{2})$	-1,7040	1,8386	-0,9953	12 (2243)
3	8	$(Obj^{2},SVs^{0,1,3},W^{0})$	-2,3199	1,7629	-0,7987	266 (4290)
	9	$(Obj^{2},SVs^{0,1,3},W^{1,4})$	-0,9571	0,2278	-0,6352	37 (2720)
	10	$(Obj^{2},SVs^{0,1,3},W^{2})$	-1,6085	1,8802	-0,5691	27 (2622)
	11	$(Obj^{2},SVs^{0,1,3},W^{3})$	-1,6458	1,9452	-0,8745	1631 (3521)

Random sets *SAM sets*

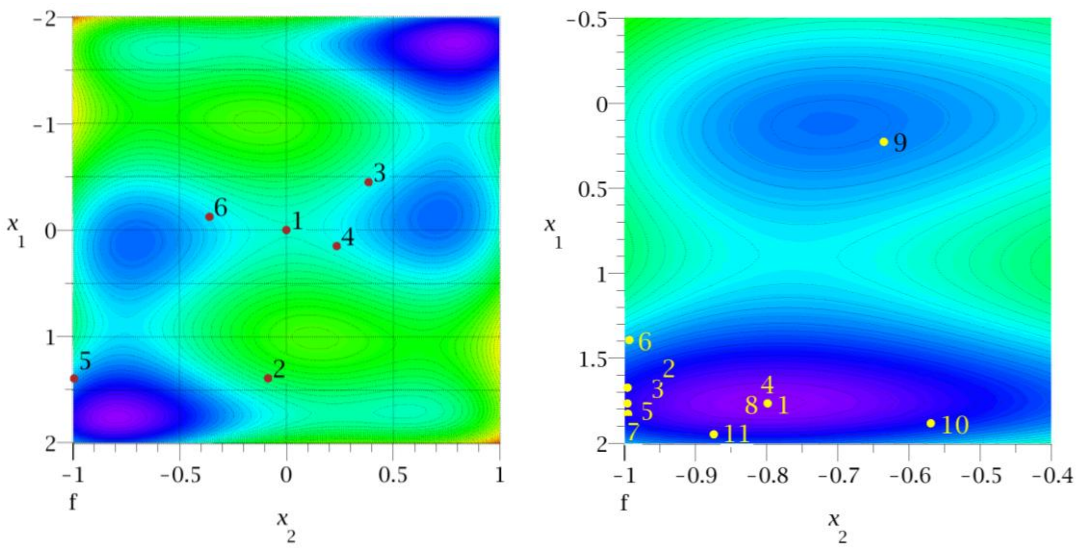

Figure 9.8 SAM analysis initiated by 5 random loops

In the first and third category where the objective function is approximated by quadratic plus cross-term and quadratic curve the optimum of the presented optimization problem is found in the cases where the weighting factor is directed at the objective function (category 1, group 4) and characteristics of obtained sets expressing their location in the design space, objective function values and feasibility/infeasibility of the sets (categories 1 and 3, groups 1 and 8). The other solutions (groups 2 and 3) including linear approximation of Obj (groups 5, 6, 7) achieved convergence near to lower constraint of the second design variable x_2. Quadratic curve fitting of the Obj and weighting directed to feasibility/infeasibility of the solution and unified weighting led the solution to converge in a local extreme (group 9). If quadratic fitting curve of the Obj is applied and weighting factor considered, especially Obj values (group 11) and distances of sets in the design space (group 10), the best sets are in ambient of the global extreme but the optimum isn't achieved.

To improve accuracy of the SAM method and possibility to approach the optimum more detailed exploration of the design space is needed. For this reason 50 random loops are performed before the SAM method is applied. The summarization of obtained results is pictured in the following table (Table 9.6) and graphically expressed in Figure 9.9.

Table 9.6 SAM analysis

Category	Group	Settings	*f*	x_1	x_2	Iterations
Initial point			*0,0000*	*0,0000*	*0,0000*	
1	1	$(\mathrm{Obj}^0,\mathrm{SVs}^{0,1,3},\mathrm{W}^0)$	-2,3199	-1,7628	0,7987	329 (3175)
	2	$(\mathrm{Obj}^0,\mathrm{SVs}^{0,1,3},\mathrm{W}^{1,4})$	-2,1143	-1,8623	0,8629	51 (2891)
	3	$(\mathrm{Obj}^0,\mathrm{SVs}^{0,1,3},\mathrm{W}^2)$	-2,3072	-1,7922	0,7985	154 (3069)
	4	$(\mathrm{Obj}^0,\mathrm{SVs}^{0,1,3},\mathrm{W}^3)$	-2,1562	-1,7567	0,6688	144 (2377)
2	5	$(\mathrm{Obj}^1,\mathrm{SVs}^{0,1,3},\mathrm{W}^0)$	-2,1143	-1,8623	0,8629	51 (3501)
	6	$(\mathrm{Obj}^1,\mathrm{SVs}^{0,1,3},\mathrm{W}^{1,4})$	-2,2075	1,7434	-0,8905	571 (2881)
	7	$(\mathrm{Obj}^1,\mathrm{SVs}^{0,1,3},\mathrm{W}^2)$	-2,1143	-1,8623	0,8629	51 (3070)
3	8	$(\mathrm{Obj}^2,\mathrm{SVs}^{0,1,3},\mathrm{W}^0)$	-2,3199	-1,7628	0,7987	320 (5256)
	9	$(\mathrm{Obj}^2,\mathrm{SVs}^{0,1,3},\mathrm{W}^{1,4})$	-2,1143	-1,8623	0,8629	51 (2891)
	10	$(\mathrm{Obj}^2,\mathrm{SVs}^{0,1,3},\mathrm{W}^2)$	-2,3199	-1,7629	0,7987	457 (2602)
	11	$(\mathrm{Obj}^2,\mathrm{SVs}^{0,1,3},\mathrm{W}^3)$	-2,1143	-1,8623	0,8629	51 (2693)

In this case the optimum was achieved in solutions where the Obj is approximated by quadratic plus cross-term and quadratic curve fitting and the weighting is directed to sets' location (group 10), their feasibility/infeasibility, Obj values and distances of design sets in the design space (groups 1 and 8).

Random sets *SAM sets*

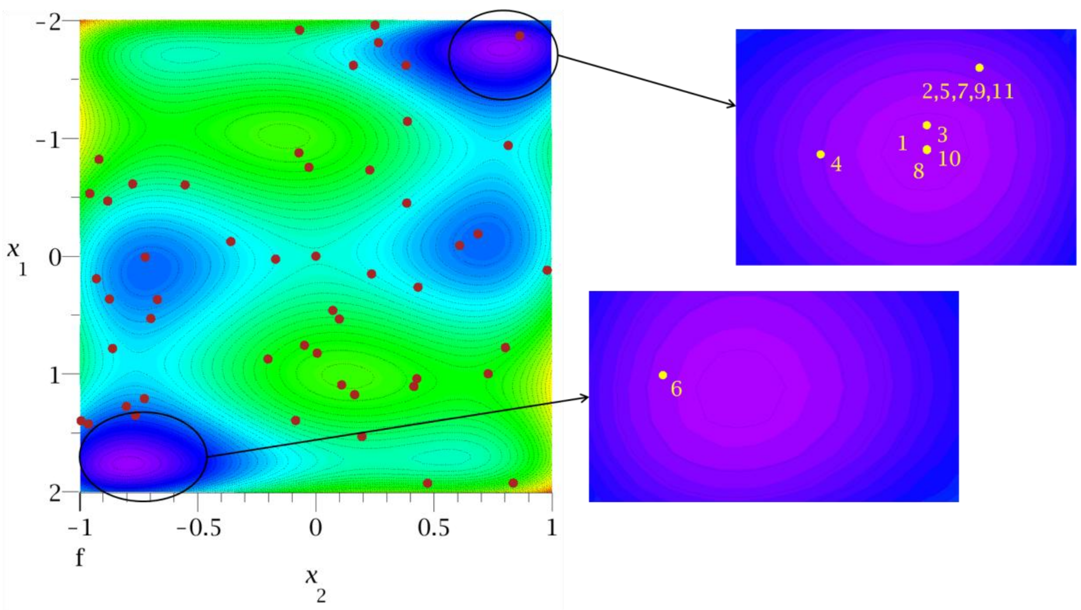

Figure 9.9 SAM analysis initiated by 50 random loops

A certain improvement occurred in the case where the weighting factor is focused on the Obj values for quadratic curve plus cross-term curve fitting (group 3) and feasibility/infeasibility of the design sets (group 4) and unified weighting of all optimization parameters for linear curve fitting (group 6). In the remaining solutions the improvement of explored design space wasn't achieved by the SAM method which means that the best design set in these cases is considered design set no. 51, which was obtained within the frame of the design space explored by the Random Tool (groups 2, 5, 7, 9, 11).

If 100 random loops are performed before the SAM method is initiated the following results are obtained (Table 9.7):

Table 9.7 SAM analysis

Category	Group	Settings	f	x_1	x_2	Iterations
Initial point			*0,0000*	*0,0000*	*0,0000*	
1	1	$(Obj^0,SVs^{0,1,3},W^0)$	-2,3146	1,7726	-0,8175	231 (2755)
	2	$(Obj^0,SVs^{0,1,3},W^{1,4})$	-2,2280	1,7482	-0,7043	68 (4619)
	3	$(Obj^0,SVs^{0,1,3},W^2)$	-2,2280	1,7482	-0,7043	68 (4365)
	4	$(Obj^0,SVs^{0,1,3},W^3)$	-2,2683	1,7899	-0,7377	158 (4354)
2	5	$(Obj^1,SVs^{0,1,3},W^0)$	-2,2280	1,7482	-0,7043	68 (2802)
	6	$(Obj^1,SVs^{0,1,3},W^{1,4})$	-2,2280	1,7482	-0,7043	68 (4594)
	7	$(Obj^1,SVs^{0,1,3},W^2)$	-2,2280	1,7482	-0,7043	68 (4509)
3	8	$(Obj^2,SVs^{0,1,3},W^0)$	-2,2280	1,7482	-0,7043	68 (4142)
	9	$(Obj^2,SVs^{0,1,3},W^{1,4})$	-2,2280	1,7482	-0,7043	68 (4602)
	10	$(Obj^2,SVs^{0,1,3},W^2)$	-2,2436	1,7382	-0,8722	258 (3919)
	11	$(Obj^2,SVs^{0,1,3},W^3)$	-2,2280	1,7482	-0,7043	68 (2909)

In this case the optimum was found only in the case where the quadratic plus cross-term curve fitting with weighting directed to the design sets' location, objective function and feasibility/infeasibility of the solution is applied (group 1). If the weighting factor is focused in Obj values and the quadratic plus cross-term curve fitting is applied (group 4) then the SAM method improves the best design set obtained by Random Tool. A similar case occurs if the Obj function is approximated by quadratic curve and weighting is directed in mutual distances of design sets in the design space (group 10). The remaining processes (groups 2, 3, 5, 6, 7, 8, 9, 11) don't improve the best random design set which is then considered as the best achieved solution of the problem.

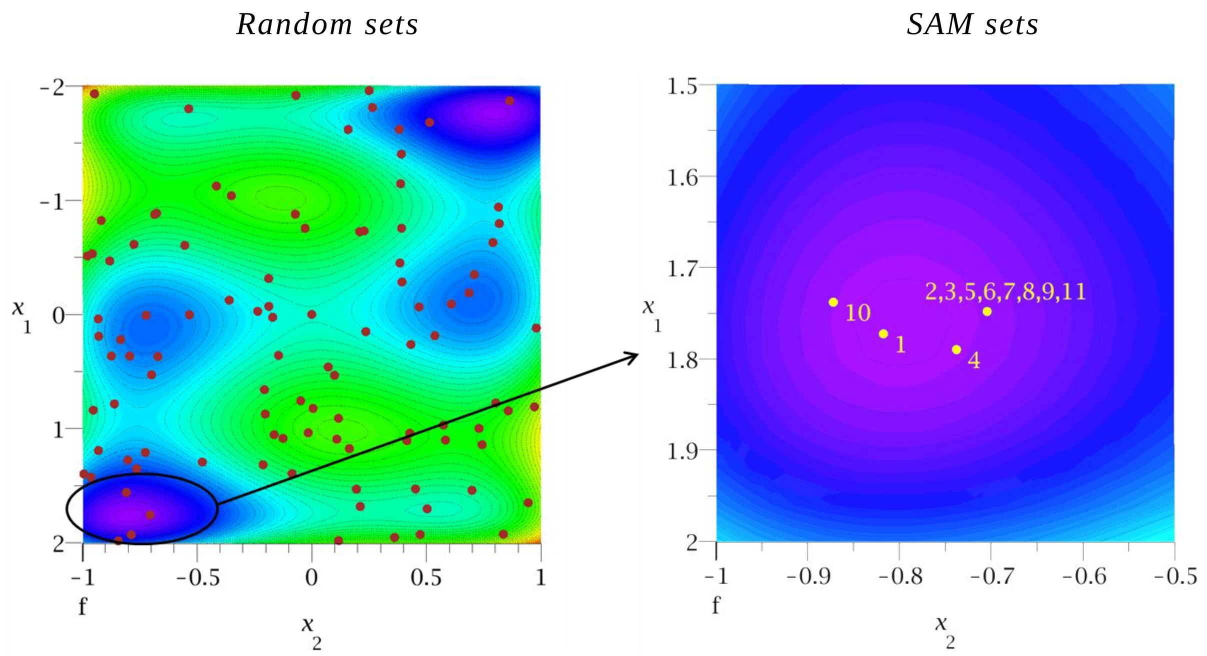

Figure 9.10 SAM analysis initiated by 100 random loops

9.1.10 Discussion and Conclusion

The performed analyses demonstrate a diversified approach in solving multi-extreme optimization problems. The accuracy and robustness of gradient (First Order Method) and approximation (Subproblem Approximation Method) method was analyzed with and without applying optimization tools, which allow a designer to explore a design space to improve features of used optimization methods. Among others the properties' specifications of the FOM method the proceeding and convergence to the optimum considerably depend on initial point definition. If features of design space are known and the aim of the problem is to specify the global extreme then the FOM method achieves satisfactory results. On the other hand if the design space shape is unknown it complicates finding global extreme and it is suitable to perform the FOM method process with varying initial points or detailed exploration of the design space by a competent tool. The SAM method proceedings are subtly influenced by an initial point definition. The SAM method represents a more general view within the frame of the design space which causes greater possibility to neglect a local extreme and finding global extreme of the objective function. Compared to the FOM method, the SAM method requires markedly more optimization loops to achieve sufficient accuracy. If high accuracy is demanded it is suitable to make competent exploration of the design space before both of the analyzed methods are applied. In the case that the designer doesn't know an estimated location of the global extreme the most general design space exploration could be achieved by the Random Tool.

Both methods achieve one of the global minimums with high accuracy corresponding to an accurate solution of the optimization problem performed in advance. To maximize robustness and accuracy of the solution, detailed exploration of the design space is needed. Generally the SAM method requires markedly more iterations than the FOM method, but on the other hand the FOM method needs more precise exploration of the design space, thus the time needed for processing might be comparable. Although features of the FOM and SAM methods and available optimization tools allow increasing global extreme localization, the robustness of the optimization process is not guaranteed.

9.2 ONE-DIMENSIONAL CONSTRAINED PROBLEM

Solving real structures' stability problems often requires considerable simplifications which are needed for analytical evaluation of the problem to verify detailed numerical model. An example of such a structure is a one end fixed beam (Figure 9.11) which represents a reinforced concrete cantilever with rectangular cross-section. The aim of the problem is to minimize its volume subject to the defined condition which is represented by vertical displacement of a free (right) end $[A]$ of the cantilever. At first the problem is solved by simple mathematical methods to achieve accurate results which are then used for results' verification obtained by First Order Method and Subproblem Approximation Method.

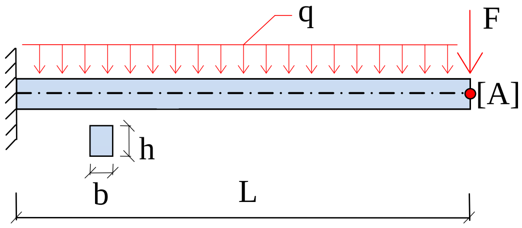

Figure 9.11 Cantilever

Optimization problems must always be defined by mathematical expressions. A general formulation of a vertical displacement at the point $[A]$ was determined by using Vereshchagin's rule. Then:

$$w_{[A]} = \frac{1}{EI}\left(\left(A^1_{M_1} T^2_{M_1}\right) + \left(A^1_{M_2} T^2_{M_2}\right)\right) \tag{9.2.1}$$

where superscript 1 denotes area of bending moment diagram in actual state and 2 indicates bending moment value in unit state in a location of a moment diagram centre of gravity in actual state. The unit state describes situation where a unit force is applied at the point $[A]$. Subscripts express load type where 1 is for distributed load q and 2 for force F. If all known constant values which do not change their values through entire optimization process are established in equation (9.2.1) the following expression is obtained:

$$w_{[A]} = \frac{1}{20.10^9.\frac{1}{12}0{,}2h}(-9000.(-\frac{9}{4}) + (-9000.(-2)))$$

then:

$$w_{[A]} = \frac{0{,}11475.10^{-3}}{h^3}. \tag{9.2.2}$$

The objective function expression which describes volume of the problem is:

$$f = b.h.l \tag{9.2.3}$$

where if known values are applied the following expression is obtained:

$$f = 0{,}2h.3$$

which is:

$$f = 0{,}6h \tag{9.2.4}$$

Then mathematical expressions of geometry and material properties of the problem which correspond to the parameters pictured in Figure 9.11 are as follows:

$$b = 0{,}2m \qquad I = 1{,}666\overline{6}.10^{-2}h^3m^4$$
$$h = \langle 0{,}1..0{,}8\rangle m \qquad E = 20.10^9 Pa$$
$$L = 3m \qquad w_{[A]} = \left[\frac{0{,}1148.10^{-3}}{h^3}\right]m$$
$$F = 2.10^3 N$$
$$q = 2.10^3 N/m \qquad f = [0{,}6h]m^3$$

According to equations (9.2.2 - 9.2.4) the optimization of the presented problem is defined in the following form:

$$\mathit{Find}\,\boldsymbol{h} = \{h\},\ \mathit{which\ minimize} \quad f(\boldsymbol{h}) = 0{,}6h \tag{9.2.5}$$

subject to

$$0{,}1 \leq h \leq 0{,}8 \tag{9.2.6}$$

$$w_{[A]} \leq 0{,}01 \tag{9.2.7}$$

where the vertical displacement value $w_{[A]}$ at the point $[A]$ is computed by the equation (9.2.2).

9.2.1 Localization of Extreme

According to the problem features, it ensued that the optimum occurs simultaneously with achieving the maximum allowed vertical displacement $w_{[A]}$ at the point $[A]$. Thus $w_{[A]} = 0{,}01m$. Then the optimal height of the cantilever is obtained by evaluation of the equation (9.2.2). It is:

$$0{,}01 = \frac{0{,}11475.10^{-3}}{h^3} \Rightarrow h = 2{,}2555.10^{-1}m$$

and the minimum value of the objective function within the frame of defined design space is:

$$f = 0{,}6 \times 0{,}2255542 = 1{,}3533.10\ m^3$$

9.2.2 Graphical Expression

According to the definition (eqs. 9.2.5 - 9.2.7) where only one design variable DV=h is defined and forms of dependent variables' functions are known (SV=$w_{[A]}$ and Obj=f) the optimization problem can be graphically expressed as it is shown in Figure 9.12. Then the optimum is achieved in an intersection of state variable expression $w_{[A]}$ (9.2.2) and its upper limit (9.2.7).

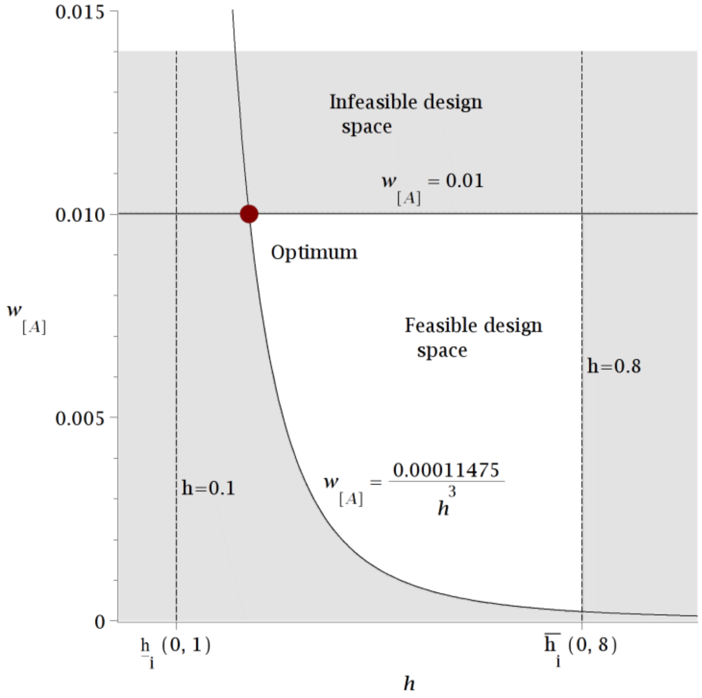

Figure 9.12 Graphical expression

In the following the problem is subjected to the Design Optimization/ANSYS module analysis where the First Order Method (FOM) and Subproblem Approximation Method (SAM) (chapter 7) are applied.

9.2.3 Solution by ANSYS/Design Optimization

A finite elements model of the presented problem was created in the Ansys program. Then the model was subjected to the optimization procedure performed by the Design Optimization module. Individual steps of the process are described below.

Parametrical model: To create the model it is necessary to parameterize quantities which are used in the optimization module to build the model and change their values (if they represent optimization variables) in each iteration. In this case there is only one design variable (DV) which represents the height h of the cantilever's cross-section. Within the frame of the Ansys program the finite elements model is created by two-dimensional elements BEAM3 where cross-sectional properties are defined by real constants. Then the cross-section (rectangle) is represented by width $b = 0{,}2m$ and height is defined by an initial value $h = 0{,}5m$ and interval $h = \langle 0{,}2; 0{,}8 \rangle m$. When the geometry and material properties are defined in the preprocessor the analysis type and boundary conditions are specified in the solution processor. Dependent variables of the problem (SVs and Obj) are evaluated and parameterized into the optimization database in the postprocessor stage. In this case they are the vertical displacement $w_{[A]}$ (as state variable SV) in the point $[A]$ and volume f (as objective function Obj) of the structure. Very important step is to verify the relevancy of the model and obtained results before the optimization procedure initiates. If there are any discrepancies in the model definition the entire optimization process could be meaningless. This is important especially in the case of complicated and complex problems where a considerable amount of time could be saved. The presented problem is verified by an accurate computational solution (see above 9.2.1).

The following procedure consists of defining parameterized quantities into the optimization database (Table 9.8), specifying an optimization tool and/or method and determining their features.

Table 9.8 Optimization variables

Variable	Expression	Description
Objective function (Obj)	f	Volume of structure
State variable (SV)	$w_{[A]}$	Vertical displacement at point $[A]$
Design variable (DV)	h	Height of cross-section

9.2.4 First Order Method

At first the First Order Method (FOM) is analyzed. Its progress is followed by different location of initial design sets and varying step lengths of gradient. The initial point of the problem is defined by DV value at first iteration. Three different values are applied; *DV=0,1; 0,5; 0,8*, where *0,1* and *0,8* correspond to lower and upper limit of DV (eq. 9.2.6). Then the FOM method is performed in each of the initial points with different step lengths, which are defined by percentage limits that are applied to the size of each line search step [75]. The percentage values limit design variables' (DVs) changes within the frame of the maximum range of the design space, which is formed by lower and upper constraints of the DVs. The optimization proceeding initiated by the initial point *DV=0,5* using the FOM method is graphically demonstrated in the figures (Figures 9.13 and 9.14) below. The Figure 9.13 shows the state variable SV (vertical displacement $w_{[A]}$ of free end of the cantilever) progressing during the FOM optimization method procedure using different step lengths. In this case convergence criteria consist of differences in objective function values (section 7.2.3).

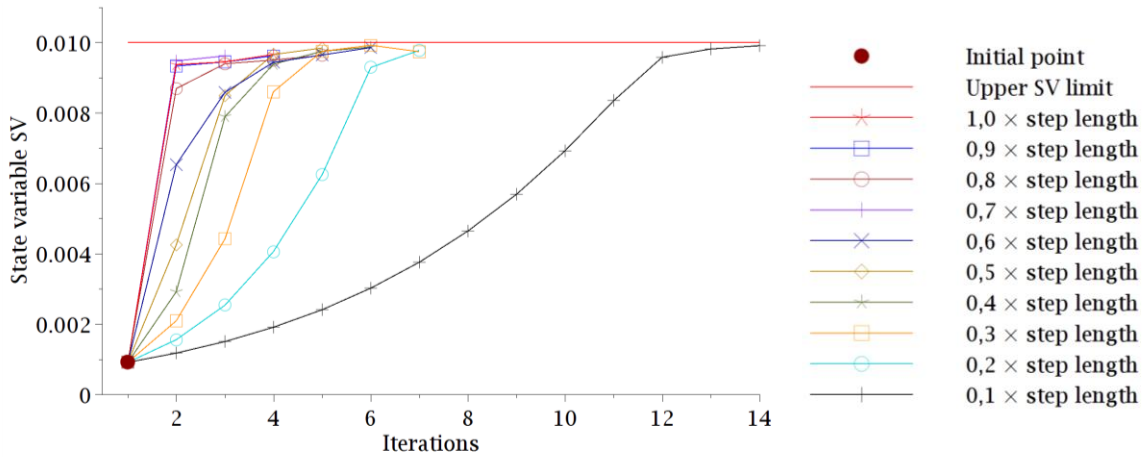

Figure 9.13 SV proceeding by FOM

The progression of the objective function Obj (volume f of the cantilever) values during the optimization procedure using different step lengths are graphically expressed in Figure 9.14.

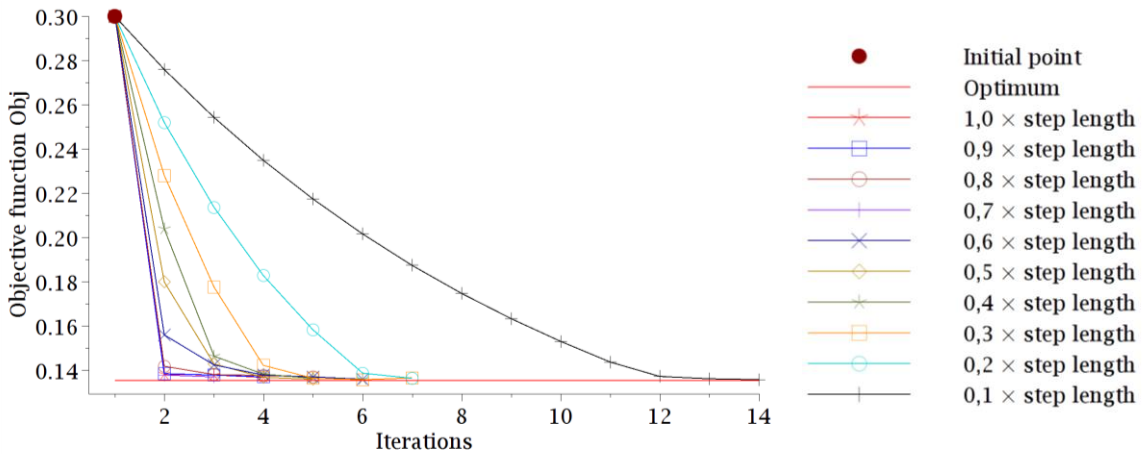

Figure 9.14 Obj proceeding by FOM

The following table (Table 9.9) illustrates optimization variables' values obtained in the best resultant design sets and total number of optimization loops performed by the FOM method for the different initial points and step lengths in the proceeding applied.

Table 9.9 FOM results

Initial point	Step length [%]	Obj.10^{-1} [m^3]	SV.10^{-3} [m]	DV.10^{-1} [m]	Iterations [-]
DV=0,1	100	1,3545	9,9730	2,2576	85 (129)
	90	1,3546	9,9726	2,2576	13 (21)
	80	1,3536	9,9947	2,2559	18 (50)
	70	1,3540	9,9845	2,2567	16 (40)
	60	1,3546	9,9726	2,2576	79 (115)
	50	1,3540	9,9857	2,2566	7 (43)
	40	1,3543	9,9778	2,2572	21 (47)
	30	1,3537	9,9913	2,2562	13 (61)
	20	1,3544	9,9772	2,2573	13 (45)
	10	1,3542	9,9817	2,2569	22 (58)
DV=0,5	100	1,3649	9,7472	2,2749	5 (8)
	90	1,3654	9,7370	2,2757	5 (8)
	80	1,3650	9,7466	2,2749	6 (9)
	70	1,3663	9,7172	2,2772	4 (7)
	60	1,3597	9,8609	2,2661	6 (10)
	50	1,3605	9,8424	2,2675	5 (9)
	40	1,3554	9,9539	2,2590	7 (9)
	30	1,3570	9,9190	2,2617	6 (11)
	20	1,3538	9,9899	2,2563	9 (11)
	10	1,3574	9,9107	2,2623	14 (18)
DV=0,8	100	1,3652	9,7422	2,2753	6 (10)
	90	1,3672	9,6991	2,2786	4 (9)
	80	1,3585	9,8864	2,2641	6 (10)
	70	1,3580	9,8980	2,2633	6 (9)
	60	1,3583	9,8899	2,2639	5 (9)
	50	1,3565	9,9302	2,2608	6 (10)
	40	1,3580	9,8970	2,2633	6 (12)
	30	1,3535	9,9961	2,2558	9 (13)
	20	1,3539	9,9870	2,2565	12 (18)
	10	1,3559	9,9427	2,2599	20 (25)

In this case where only one DV and two dependent variables (SV and Obj) are defined and the features of the problem represent a strictly convex optimization problem the FOM method achieves satisfying results. All obtained best design sets are in ambient of the optimum. Deviations of obtained objective function values are under 1% against the accurate solution (section 9.2.1).

9.2.5 Subproblem Approximation Method

Subproblem Approximation Method is based on approximated dependent functions SVs and Obj (section 7.1). The method allows determining an importance of optimization variables due to weight factors within the frame of the least squares method approximation. The least squares method approximation (fitting) can be performed by linear, quadratic or quadratic function with cross-term form (section 7.1.1). The SAM method requires at least four points (design sets) to initiate the approximation of dependent variables. These can be performed by any available optimization tool (section 7.3), or if the tool is not defined by a designer then 4 random design sets are performed by default. In the case of presented optimization problem the objective function approximation doesn't change its form during curve fitting because it is defined by a linear equation (9.2.4). The state variable is defined by a quadratic equation (9.2.2) which leads to the same results obtained by quadratic and quadratic plus cross-term fitting approximation. The features of individual solutions are marked by superscripts (their explanation is in Table 9.4) assigned to the appropriate variable. In the following the results in the SAM analysis obtained are summarized. The analyses were initiated by 5 loops (Table 9.10), 50 loops (Table 9.11) and 100 loops (Table 9.12) performed by the Random Tool (section 7.3.1).

Table 9.10 SAM results initiated by 5 random loops

Category	Group	Settings	Obj.10^{-1} [m^3]	SV.10^{-3} [m]	DV.10^{-1} [m]	Iterations [-]
1	1	($Obj^{0,1,2}$,$SVs^{0,3}$,W^0)	1,3534	9,9985	2,2556	22 (153)
	2	($Obj^{0,1,2}$,$SVs^{0,3}$,W^1)	1,3545	9,9737	2,2575	120 (1171)
	3	($Obj^{0,1,2}$,$SVs^{0,3}$,W^2)	1,3560	9,9403	2,2601	125 (1171)
	4	($Obj^{0,1,2}$,$SVs^{0,3}$,W^3)	1,3564	9,3150	2,2607	128 (1169)
	5	($Obj^{0,1,2}$,$SVs^{0,3}$,W^4)	1,3533	9,9998	2,2556	169 (1191)
2	6	($Obj^{0,1,2}$,SVs^1,W^0)	1,3534	9,9994	2,2556	51 (1184)
	7	($Obj^{0,1,2}$,SVs^1,W^1)	1,3534	9,9981	2,2557	22 (1177)
	8	($Obj^{0,1,2}$,SVs^1,W^2)	1,3541	9,9829	2,2568	129 (1170)
	9	($Obj^{0,1,2}$,SVs^1,W^3)	1,3543	9,9778	2,2572	127 (1171)
	10	($Obj^{0,1,2}$,SVs^1,W^4)	1,4375	8,3441	2,3958	12 (1170)

The following figures (Figure 9.15 and 9.16) show the proceedings in the first 20 loops by the SAM method performed (initiated by 5 random loops) depending on different settings of variables’ weights and dependent variables' curve fitting within the frame of the presented problem. The Figure 9.15 represents the objective function values' proceedings and the Figure 9.16 shows the progress of state variable's values.

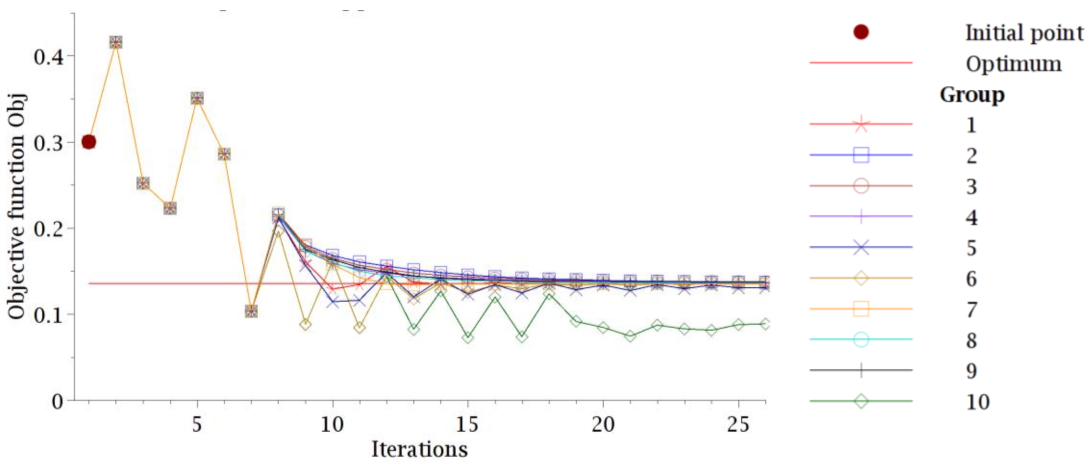

Figure 9.15 Obj proceeding by SAM

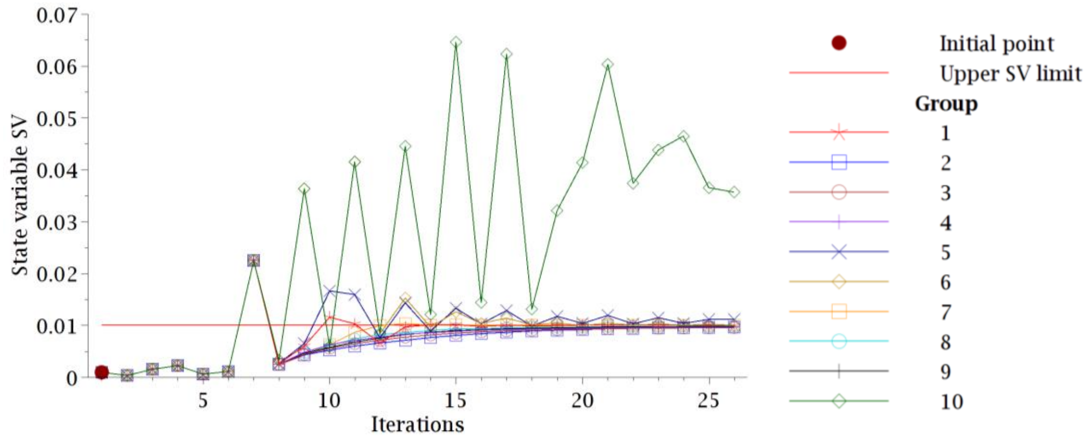

Figure 9.16 SV proceeding by SAM

Table 9.11 SAM results initiated by 50 random loops

Category	Group	Settings	Obj.10^{-1} [m^3]	SV.10^{-3} [m]	DV.10^{-1} [m]	Iterations [-]
1	1	($Obj^{0,1,2}$,$SVs^{0,3}$,W^0)	1,3534	9,9987	2,2556	192 (1237)
	2	($Obj^{0,1,2}$,$SVs^{0,3}$,W^1)	1,3728	9,5809	2,2880	142 (1216)
	3	($Obj^{0,1,2}$,$SVs^{0,3}$,W^2)	1,3548	9,9663	2,2581	144 (1217)
	4	($Obj^{0,1,2}$,$SVs^{0,3}$,W^3)	1,3575	9,9088	2,2624	139 (1212)
	5	($Obj^{0,1,2}$,$SVs^{0,3}$,W^4)	1,3686	9,6679	2,2811	59 (1217)
2	6	($Obj^{0,1,2}$,SVs^1,W^0)	1,3534	9,9994	2,2556	116 (1233)
	7	($Obj^{0,1,2}$,SVs^1,W^1)	1,5092	7,2105	2,5153	134 (1219)
	8	($Obj^{0,1,2}$,SVs^1,W^2)	1,3784	9,4651	2,2973	136 (1210)
	9	($Obj^{0,1,2}$,SVs^1,W^3)	1,3552	9,9595	2,2586	142 (1214)
	10	($Obj^{0,1,2}$,SVs^1,W^4)	1,4808	7,6327	2,4681	55 (1213)

In the cases where the solution is initiated by 5 and 50 random loops the SAM proceeding improves the best sets obtained by the Random Tool. If the weight factors are directed into feasibility/infeasibility of obtained design sets or are unified and approximation of SV is performed by linear fitting, the solutions approach the optimum ambient with difficulties. The remaining solutions achieve the optimum with admissible accuracy.

Table 9.12 SAM results initiated by 100 random loops

Category	Group	Settings	Obj.10^{-1} [m^3]	SV.10^{-3} [m]	DV.10^{-1} [m]	Iterations [-]
1	1	($Obj^{0,1,2}$,$SVs^{0,3}$,W^0)	1,3540	9,9856	2,2566	115 (1267)
	2	($Obj^{0,1,2}$,$SVs^{0,3}$,W^1)	1,5033	7,2964	2,5054	80 (1274)
	3	($Obj^{0,1,2}$,$SVs^{0,3}$,W^2)	1,3623	9,8030	2,2705	145 (1260)
	4	($Obj^{0,1,2}$,$SVs^{0,3}$,W^3)	1,3572	9,9143	2,2620	146 (1259)
	5	($Obj^{0,1,2}$,$SVs^{0,3}$,W^4)	1,3654	9,7368	2,2757	103 (1261)
2	6	($Obj^{0,1,2}$,SVs^1,W^0)	1,3610	9,8320	2,2683	107 (1264)
	7	($Obj^{0,1,2}$,SVs^1,W^1)	1,5020	7,3154	2,5033	141 (1277)
	8	($Obj^{0,1,2}$,SVs^1,W^2)	1,4610	7,9472	2,4351	147 (1274)
	9	($Obj^{0,1,2}$,SVs^1,W^3)	1,5033	7,2964	2,5054	80 (1273)
	10	($Obj^{0,1,2}$,SVs^1,W^4)	1,3577	9,9035	2,2628	103 (1261)

If 100 random loops are performed the design set number 80 is already located quite near the optimum. In the cases where the weight factor lays stress on the objective function values or the weights are unified the SAM method isn't able to improve the best random design set. In the other cases the best random design set was improved by the SAM method and the convergence occurred in ambient of the optimum with convenient tolerance.

9.2.6 Discussion and Conclusion

Within the frame of the presented strictly convex optimization problem which is expressed by the FEM model of cantilever, the First Order Method and Subproblem Approximation Method were analyzed.

The strictly convex features of the problem lead to the optimum in all analyzed step lengths of the First Order Method considering three different initial points. The differences in achieved results consist in proceeding time. Generally the solution with the shortest step length definition might be considered as the most time consuming. On the other hand short steps lead slowly to the optimum by only one direction. Long step lengths might cause over-jumping the optimum and the solution continues from the other side than the initial point is defined. In this case the solution could require more iteration to achieve the actual optimum. If the convexity of the problem is known but not the current position of the optimum, short step length is sufficient to achieve the optimum with high accuracy.

The Subproblem Approximation Method was analyzed with varying weighting factor features and approximation of dependent variables. It was applied in the cases where the design space was explored by 5, 50 and 100 loops performed by the Random Tool in advance. The objective function of the problem is defined by a linear equation which causes using linear fitting approximation only. The state variable SV is expressed by a quadratic equation which leads to the same results obtained by quadratic and quadratic plus cross-term approximations. The manner of SV approximation doesn't play a big role in the presented problem. The weighting factor has the greatest effect. The achieved results showed that the most efficient in this case is directing the weighting factor to the design space, objective function and feasibility/infeasibility of the solution.

9.3 TWO-DIMENSIONAL CONSTRAINED PROBLEM

The aim of the following optimization problem is to minimize the volume of a two-bar plane frame structure (Figure 9.17) subjected to vertical displacement limit w of the structure in the point $[A]$. This can be acquired by varying of bars' cross-sections' heights defined by parameters h_1 and h_2. The structure is fixed at the bottom of the column which is also loaded by single cross force F. The horizontal bar is subjugated by distributed load q.

At first the problem is analyzed by graphical and manual solution to achieve an optimum, which is then used for results' verification acquired by First Order Method and Subproblem Approximation method.

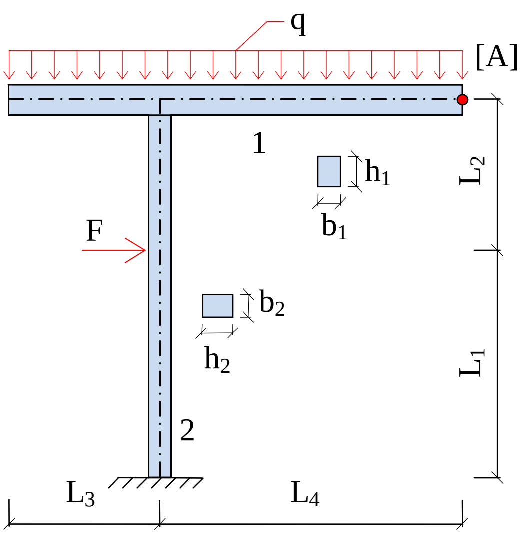

Figure 9.17 Two-bar plane frame structure

Vereshchagin's rule was used to determine vertical displacement at the point $[A]$ in a mathematical form. Then:

$$w_{[A]} = \frac{1}{EI_1}\left(\left(A^1_{M_{1-2}} T^2_{M_{1-2}}\right) + \left(A^1_{M_1} T^2_{M_1}\right)\right) + \frac{1}{EI_2}\left(A^1_{M_4} T^2_{M_4}\right). \tag{9.3.1}$$

After substitution of known constant values we obtain:

$$w_{[A]} = \frac{1}{20.10^9.\frac{1}{12}0{,}2h_2^3}(240000 + 360000) + \frac{1}{20.10^9.\frac{1}{12}0{,}2h_1^3}\left(-\frac{64000}{3}.(-3)\right)$$

and then:

$$w_{[A]} = \frac{1{,}799\overline{9}.10^{-3}}{h_2^3} + \frac{1{,}919\overline{9}.10^{-4}}{h_1^3} \tag{9.3.2}$$

The volume of the structure is stated as follows:

$$f = (b_1 h_1 (l_1 + l_2)) + (b_2 h_2 (l_3 + l_4)) \qquad (9.3.3)$$

then:

$$f = (0{,}2h_1.5) + (0{,}2h_2.6)$$

which is:

$$f = h_1 + 1{,}2h_2 \qquad (9.3.4)$$

Mathematical expressions of geometry and material properties of the problem according to parameters pictured in Figure 9.17 are as follows:

$$\begin{array}{ll}
b_1 = 0{,}2m & I_1 = 1{,}666\overline{6}.10^{-2} h_1^3 m^4 \\
b_2 = 0{,}2m & I_2 = 1{,}666\overline{6}.10^{-2} h_2^3 m^4 \\
h_1 = \langle 0{,}1..0{,}8 \rangle m & E = 20.10^9 Pa \\
h_2 = \langle 0{,}1..0{,}8 \rangle m & F = 20.10^3 N \\
L_1 = 3m & q = 2.10^3 N/m \\
L_2 = 2m & \\
L_3 = 2m & w_{[A]} = \left[\dfrac{1{,}7999.10^{-3}}{h_2^3} + \dfrac{1{,}9199.10^{-4}}{h_1^3} \right] m \\
 & f = [1{,}0h_1 + 1{,}2h_2] m^3
\end{array}$$

Then the optimization problem is defined:

$$Find\, \boldsymbol{h} = \begin{Bmatrix} h_1 \\ h_2 \end{Bmatrix}, which\ minimize \quad f(\boldsymbol{h}) = 1{,}0h_1 + 1{,}2h_2 \qquad (9.3.5)$$

subject to:

$$\begin{array}{c} 0{,}1 \le h_1 \le 0{,}8 \\ 0{,}1 \le h_2 \le 0{,}8 \end{array} \qquad (9.3.6)$$

$$w_{[A]} \le 0{,}02 \qquad (9.3.7)$$

where $w_{[A]}$ is expressed by the equation (9.3.2).

9.3.1 Graphical Expression

According to the definition (eqs. 9.3.5 – 9.3.7) a graphical expression of the presented problem can be obtained (Figure 9.18). The axes of the graphical expression are represented by independent variables (DVs) and the feasible space is originated by an intersection of lower and upper limits of design variables DVs h_1, h_2 and upper limit of state variable SV $w_{[A]}$. The optimum of the optimization problem is located at the point where the objective function forms a tangent to the state variable function. The actual optimization problem represents a strictly convex optimization problem which leads to an existence of one and only one extreme within the frame of the defined design space.

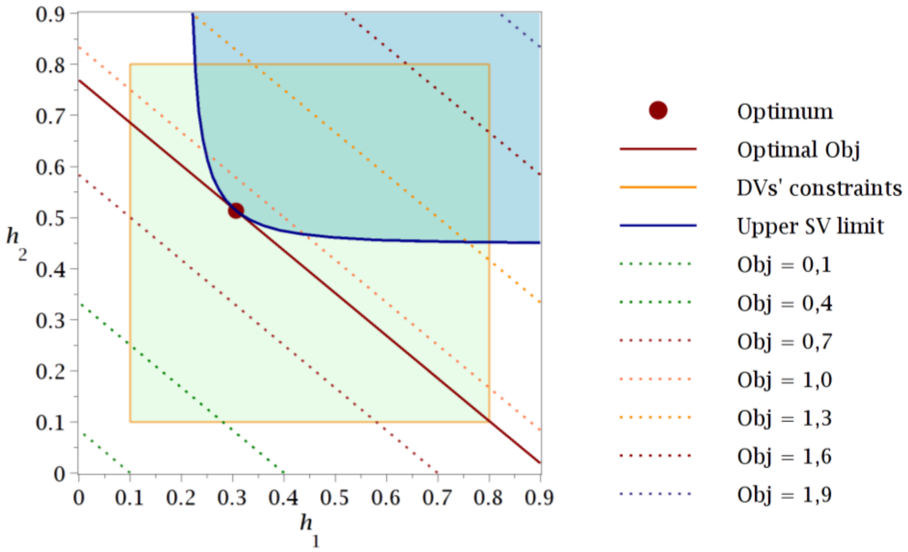

Figure 9.18 Graphical expression

9.3.2 Localization of Extreme

By estimation obtained from the graphical expression, the design variables' (DVs) intervals are defined as follows:

$$h_1 = \langle 0{,}25 \ldots 0{,}35\rangle$$
$$h_2 = \langle 0{,}45 \ldots 0{,}55\rangle^{\cdot}$$

To find the optimum of the problem a bisection method was used. Then the optimal point of the problem is defined by DVs' values:

$$h_1 = 3{,}0674.10^{-1}\ m$$
$$h_2 = 5{,}1282.10^{-1}\ m^{\cdot}$$

According to DVs' values the state variable (SV) and objective function (Obj) values are:

$$w_{[A]} = 0{,}02\ m$$

$$f = 9{,}2215.10^{-1}\ m^3.$$

9.3.3 Solution by ANSYS/Design Optimization

Before an optimization procedure is performed, a finite elements model and a parametrical model of the actual problem is created within the frame of the Ansys program. The FEM/FEA model is assembled from two-dimensional elements BEAM3. Then the optimization variables (parameters) are defined as follows: The bars' cross-sections' heights h_1 and h_2 represent independent design variables (DVs) and dependent variables are expressed by vertical displacement at the point $[A]$ as a state variable (SV) and weight f of the structure as the objective function (Obj). The parameters defined in this problem are summarized in the following table (Table 9.13).

Table 9.13 Optimization variables

Variable	Expression	Description
(Obj)	f	Volume of structure
(SV)	$w_{[A]}$	Vertical displacement at point $[A]$
(DV_1)	h_1	Height of horizontal bar (1) cross-section
(DV_2)	h_2	Height of column (2) cross-section

9.3.4 First Order Method

To analyze the efficiency of the FOM method different cases with varying step lengths' range of gradients and different initial point location are applied. The following table (Table 9.14) shows optimization variables' values at analyzed initial points,

Table 9.14 Initial design sets

Obj $[m^3]$	SV [m]	DV_1 [m]	DV_2 [m]	Status
0,2200	1,9922	0,1	0,1	infeasible
1,1000	$1{,}5966.10^{-2}$	0,5	0,5	feasible
1,7600	$3{,}9094.10^{-3}$	0,8	0,8	feasible

where the initial points $DV_1=DV_2=0{,}5$ and $DV_1=DV_2=0{,}8$ are located in the feasible design space and $DV_1=DV_2=0{,}1$ is in the infeasible design space because of exceeding the upper SV limit (9.3.7).

The progress of the dependent variables (objective function Obj and state variable SV) due 10 loops by the FOM method performed initiated by $h_1=0{,}5$ and $h_2=0{,}5$ is shown in the figures (Figure 9.19 and 9.20) below.

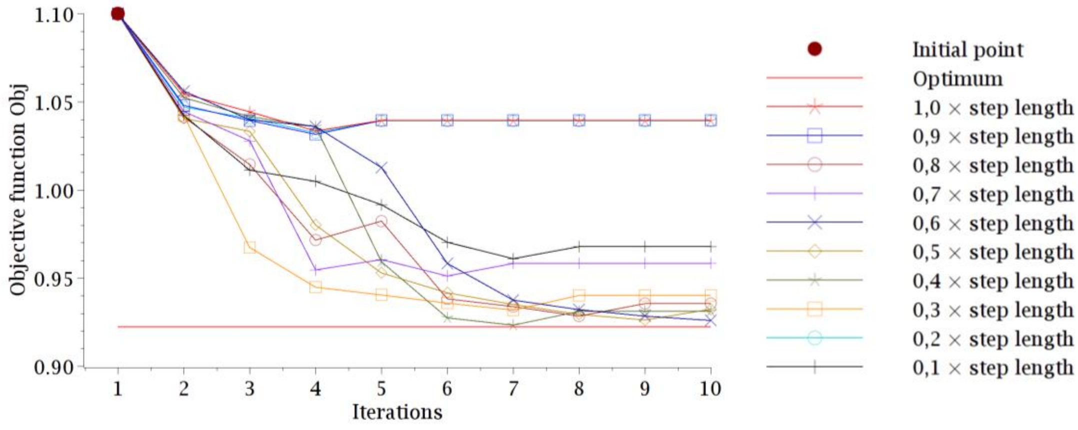

Figure 9.19 Obj proceeding by FOM

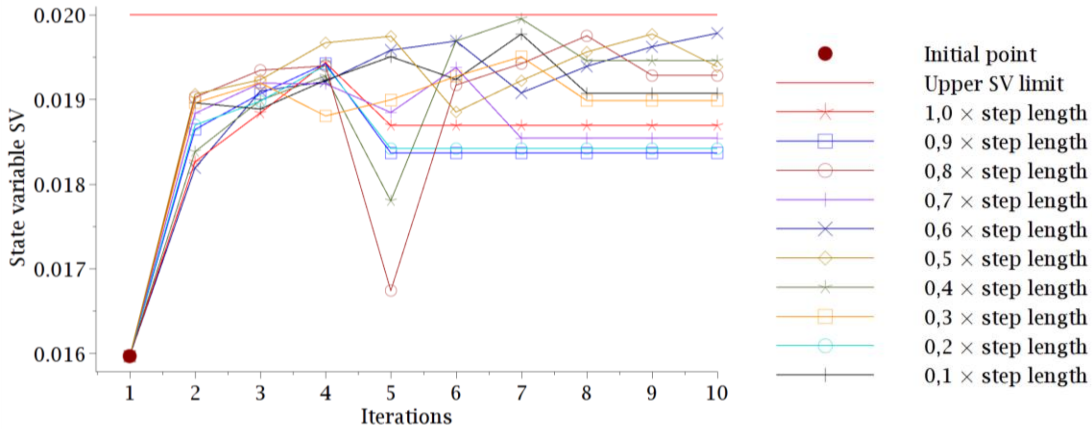

Figure 9.20 SV proceeding by FOM

Termination of the procedure consists in achieving of convergence criteria which are in this case defined as objective function values' differences in two consecutive design sets (section 7.2.3).

The variables' values in the best sets obtained and the total number of iterations by the FOM method performed are summarized in Table 9.15.

Table 9.15 FOM results

Initial point	**Step length [%]**	**Obj.10^{-1} [m^3]**	**SV.10^{-2} [m]**	**DV_1.10^{-1} [m]**	**DV_2.10^{-1} [m]**	**Iterations [-]**
DV_1=0,1 DV_2=0,1	100	9,2369	1,9927	3,0688	5,1401	39 (100)
	90	9,2378	1,9921	3,0691	5,1406	43 (103)
	80	9,2316	1,9962	3,0631	5,1657	52 (107)
	70	9,2357	1,9942	3,1058	5,1083	16 (102)
	60	9,2301	1,9971	3,0665	5,1363	43 (104)
	50	9,2300	1,9972	3,0751	5,1291	46 (103)
	40	9,2307	1,9967	3,0646	5,1384	93 (129)
	30	9,2313	1,9967	3,0456	5,1547	29 (102)
	20	9,2315	1,9967	3,1000	5,1096	47 (101)
	10	9,2278	1,9986	3,0632	5,1372	99 (129)
DV_1=0,5 DV_2=0,5	100	10,3350	1,9435	4,7214	4,6784	4 (9)
	90	10,3160	1,9422	4,6972	4,6821	4 (9)
	80	9,2844	1,9750	3,2340	5,0420	8 (12)
	70	9,5122	1,9376	3,6409	4,8928	6 (10)
	60	9,2593	1,9783	3,0707	5,1572	10 (13)
	50	9,2613	1,9770	3,0829	5,1486	9 (12)
	40	9,2330	1,9954	3,0887	5,1203	7 (12)
	30	9,3188	1,9503	3,2277	5,0759	7 (12)
	20	10,3250	1,9403	4,7054	4,6829	4 (9)
	10	9,6089	1,9771	3,5651	4,7865	7 (12)
DV_1=0,8 DV_2=0,8	100	9,4574	1,9827	3,6431	4,8452	8 (13)
	90	10,6850	1,9491	5,1173	4,6394	4 (9)
	80	9,2676	1,9871	2,9349	5,2773	7 (12)
	70	9,3208	1,9496	2,9724	5,2903	10 (16)
	60	10,7110	1,9326	5,1280	4,6528	4 (10)
	50	9,4097	1,9442	3,4624	4,9561	7 (12)
	40	10,6520	1,9657	5,1005	4,6265	6 (11)
	30	9,4752	1,9846	3,6763	4,8325	7 (14)
	20	9,3090	1,9881	3,3653	4,9531	12 (15)
	10	10,2030	1,9932	4,6296	4,6446	10 (15)

In the case where the FOM method is initiated in the infeasible design space the solution requires markedly more iterations to achieve convergence criteria. On the other hand the high number of performed loops allows finding the minimum of the objective function in the feasible design space. The FOM analyses initiated by the feasible design sets require less number of loops to achieve convergence criteria at the expense of accuracy. To guarantee achieving the optimum of the problem, design space exploration is recommended. In the case where the presented problem is initiated by 5 random loops the optimum is obtained in all presented cases.

9.3.5 Subproblem Approximation Method

The Subproblem Approximation Method was analyzed according to different approximation of dependent variables SV and Obj and different pointing of weighting factor (Table 9.4). The objective function in this problem is represented by the linear equation (9.3.4) which leads to its changeless form due the different approximation proceeding. Furthermore the SAM method is analyzed depending on different number of random iterations which are evaluated before the first SAM approximation is formed. The design space was explored by 5, 50 and 100 loops where the Random Tool (7.3.1) was applied. The following table (Table 9.16) represents summarized variables' values in the best design sets obtained in the analysis where the SAM proceeding was initiated by 5 random loops. The presented results are divided into three categories according to approximation type of dependent variable SV.

Table 9.16 SAM results initiated by 5 random loops

Category	**Group**	**Settings**	**Obj.10^{-1} [m^3]**	**SV.10^{-2} [m]**	**DV_1.10^{-1} [m]**	**DV_2.10^{-1} [m]**	**Iterations [-]**
1	1	$(Obj^{0,1,2},SVs^0,W^0)$	9,2274	1,9988	3,0672	5,1335	147 (1219)
	2	$(Obj^{0,1,2},SVs^0,W^1)$	9,2273	1,9990	3,0620	5,1377	398 (1374)
	3	$(Obj^{0,1,2},SVs^0,W^2)$	9,2278	1,9986	3,0719	5,1299	381 (1257)
	4	$(Obj^{0,1,2},SVs^0,W^3)$	9,6820	1,9892	2,5796	5,9187	27 (1176)
	5	$(Obj^{0,1,2},SVs^0,W^4)$	9,2291	1,9978	3,0670	5,1351	150 (1177)
2	6	$(Obj^{0,1,2},SVs^1,W^{0,4})$	10,0170	1,8130	2,6488	6,1404	10 (1168)
	7	$(Obj^{0,1,2},SVs^1,W^1)$	9,7336	1,7053	3,1723	5,4678	8 (1166)
	8	$(Obj^{0,1,2},SVs^1,W^2)$	9,3746	1,9658	2,8143	5,4669	14 (1228)
	9	$(Obj^{0,1,2},SVs^1,W^3)$	10,2580	1,8406	2,5490	6,4241	18 (1183)
3	10	$(Obj^{0,1,2},SVs^3,W^0)$	9,2278	1,9986	3,0702	5,1313	135 (1217)
	11	$(Obj^{0,1,2},SVs^3,W^1)$	9,2290	1,9979	3,0595	5,1413	337 (1219)
	12	$(Obj^{0,1,2},SVs^3,W^2)$	9,2300	1,9986	3,0224	5,1730	158 (1189)
	13	$(Obj^{0,1,2},SVs^3,W^3)$	9,2607	1,9783	3,0424	5,1820	134 (1181)
	14	$(Obj^{0,1,2},SVs^3,W^4)$	9,2281	1,9984	3,0774	5,1256	476 (1458)

In the case where the SAM method proceeding is initiated by 5 random loops, the SV approximation performed by linear fitting prevents accuracy of the solution. Also, quadratic SV approximation with combination of weight factors directed into the objective function values converges at a distant location from the actual optimum. The remaining cases where the quadratic and quadratic plus cross-term fitting approximate the SV variable achieve ambient of the optimum.

Progress of dependent variables (Obj and SV) due to the first 30 SAM iterations initiated by the 5 random loops is pictured in the following figures (Figure 9.21 and 9.22).

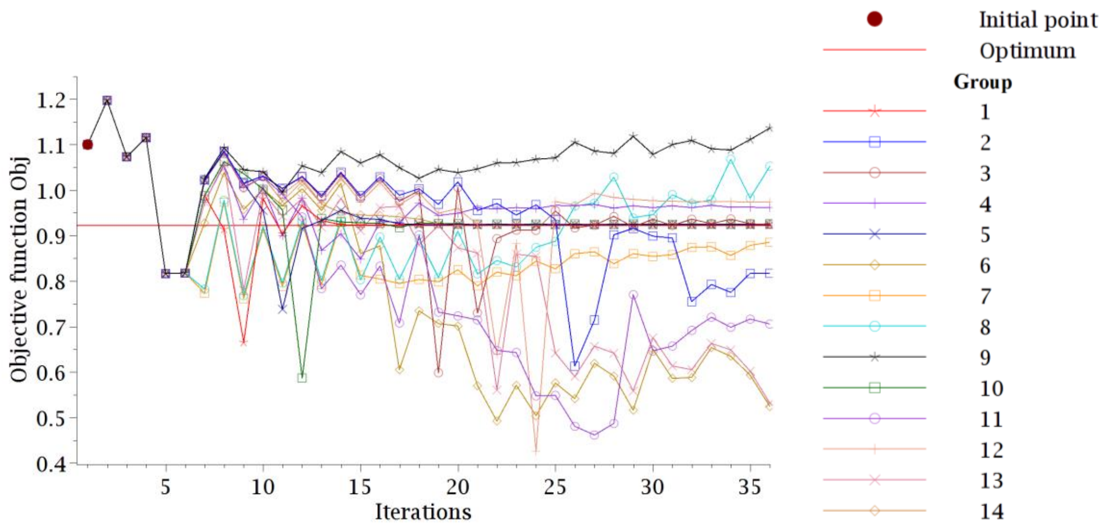

Figure 9.21 Obj proceeding by SAM

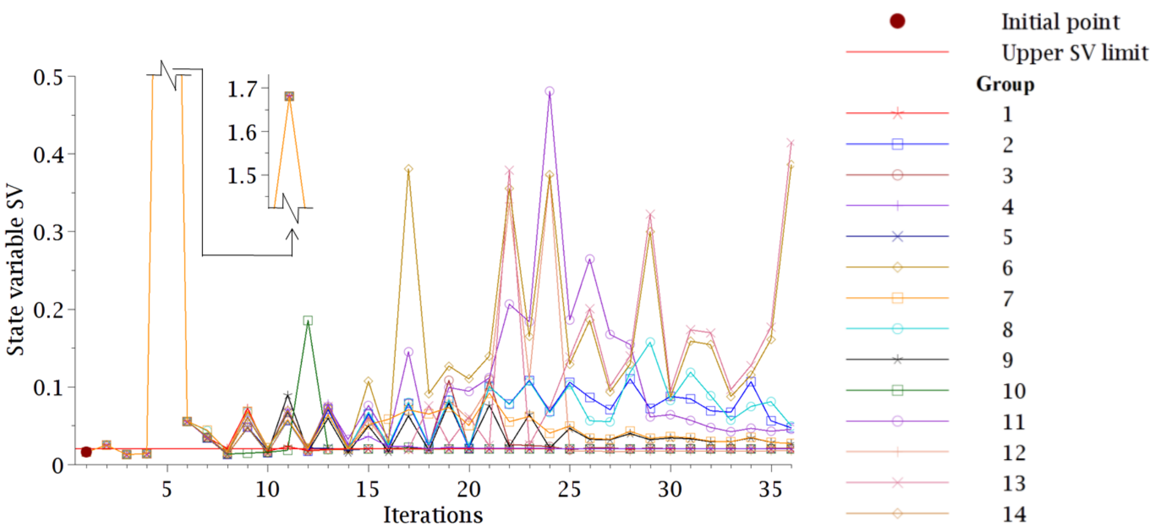

Figure 9.22 SV proceeding by SAM

Features of the best design sets and numbers of iterations by the SAM method performed initiated by 50 and 100 loops are summarized in the following tables (Table 9.18 and 9.19).

Table 9.17 SAM results initiated by 50 random loops

Category	Group	Settings	Obj.10^{-1} [m^3]	SV.10^{-2} [m]	$DV_1.10^{-1}$ [m]	$DV_2.10^{-1}$ [m]	Iterations [-]
1	1	$(Obj^{0,1,2},SVs^0,W^0)$	9,2286	1,9981	3,0771	5,1262	177 (1211)
	2	$(Obj^{0,1,2},SVs^0,W^1)$	9,3479	1,9959	2,7764	5,4763	176 (1256)
	3	$(Obj^{0,1,2},SVs^0,W^2)$	9,2556	1,9900	2,9584	5,2477	169 (1366)
	4	$(Obj^{0,1,2},SVs^0,W^3)$	9,2340	1,9946	3,0631	5,1424	158 (1242)
	5	$(Obj^{0,1,2},SVs^0,W^4)$	9,3712	1,9953	2,7542	5,5142	57 (1255)
2	6	$(Obj^{0,1,2},SVs^1,W^0)$	9,5349	1,8691	2,8606	5,5619	152 (1229)
	7	$(Obj^{0,1,2},SVs^1,W^1)$	9,5854	1,7841	3,2265	5,2991	19 (1311)
	8	$(Obj^{0,1,2},SVs^1,W^2)$	9,4194	1,9669	2,7642	5,5459	53 (1211)
	9	$(Obj^{0,1,2},SVs^1,W^3)$	9,5854	1,7841	3,2265	5,2991	19 (1216)
	10	$(Obj^{0,1,2},SVs^1,W^4)$	9,4098	1,9723	2,7626	5,5393	137 (1216)
3	11	$(Obj^{0,1,2},SVs^3,W^0)$	9,2326	1,9988	3,1436	5,0742	143 (1221)
	12	$(Obj^{0,1,2},SVs^3,W^1)$	9,3242	1,9977	2,7993	5,4374	252 (1221)
	13	$(Obj^{0,1,2},SVs^3,W^2)$	9,2370	1,9974	2,9866	5,2086	158 (1234)
	14	$(Obj^{0,1,2},SVs^3,W^3)$	9,2362	1,9988	2,9787	5,2146	346 (1259)
	15	$(Obj^{0,1,2},SVs^3,W^4)$	9,3593	1,9935	2,7696	5,4914	194 (1243)

Exploration of the design space by 50 random loops causes more reliable convergence to the optimum at the expense of accuracy. Except in linear approximation of the SV, the ambient of the actual optimum was found.

Table 9.18 SAM results initiated by 100 random loops

Category	Group	Settings	Obj.10^{-1} [m^3]	SV.10^{-2} [m]	$DV_1.10^{-1}$ [m]	$DV_2.10^{-1}$ [m]	Iterations [-]
1	1	$(Obj^{0,1,2},SVs^0,W^0)$	9,2291	1,9978	3,0670	5,1351	243 (1263)
	2	$(Obj^{0,1,2},SVs^0,W^1)$	9,2346	1,9944	3,0527	5,1516	249 (1333)
	3	$(Obj^{0,1,2},SVs^0,W^2)$	9,2791	1,9685	3,0188	5,2169	194 (1319)
	4	$(Obj^{0,1,2},SVs^0,W^3)$	9,2313	1,9964	3,0607	5,1421	197 (1271)
	5	$(Obj^{0,1,2},SVs^0,W^4)$	9,4314	1,9984	2,6992	5,6102	839 (1278)
2	6	$(Obj^{0,1,2},SVs^1,W^0)$	9,5152	1,8259	3,2377	5,2312	84 (1259)
	7	$(Obj^{0,1,2},SVs^1,W^1)$	9,5152	1,8259	3,2377	5,2312	84 (1302)
	8	$(Obj^{0,1,2},SVs^1,W^2)$	9,4516	1,9916	2,6955	5,6302	243 (1326)
	9	$(Obj^{0,1,2},SVs^1,W^3)$	9,5152	1,8259	3,2377	5,2312	84 (1295)
	10	$(Obj^{0,1,2},SVs^1,W^4)$	9,5152	1,8259	3,2377	5,2312	84 (1259)
3	11	$(Obj^{0,1,2},SVs^3,W^0)$	9,2292	1,9977	3,0678	5,1345	379 (1334)
	12	$(Obj^{0,1,2},SVs^3,W^1)$	9,3157	1,9980	2,8093	5,4220	247 (1276)
	13	$(Obj^{0,1,2},SVs^3,W^2)$	9,2391	1,9951	2,9960	5,2026	255 (1429)
	14	$(Obj^{0,1,2},SVs^3,W^3)$	9,2333	1,9987	2,9954	5,1983	246 (1309)
	15	$(Obj^{0,1,2},SVs^3,W^4)$	9,3926	1,9854	2,7532	5,5329	987 (1298)

If the solution is initiated by 100 random loops the SAM method requires a great number of iterations to achieve convergence especially if weighting factor is directed to feasibility/infeasibility of obtained design sets. In the case that the SV variable is approximated by a linear fitting the SAM method isn't able to improve features of the best achieved random design set (no. 84). The actual optimum ambient is achieved in solutions where the SV approximation is performed by quadratic and quadratic plus cross-term fitting.

9.3.6 Discussion and Conclusion

First Order Method and Subproblem Approximation Method were analyzed within the frame of a two-bar plane frame structure weight minimization. The optimization problem is expressed by two dependent (Obj and SV) and two independent variables (DVs). To control and verify efficiency and accuracy of analyzed methods' resultant design sets, the accurate manually evaluated and graphical solution was performed.

The First Order Method was analyzed depending on different initial point location and varying step lengths of gradient. The initial points were located at the lower and upper constraints of the design variables and in the middle of their defined ranges. In the case where the initial point is defined in the infeasible design space (lower constraints of the DVs) the FOM method requires markedly more iterations to achieve convergence criteria, but with no effect on efficiency and accuracy of the solution. If the initial point is localized in the feasible design space the convergence criteria were achieved already in max 10 iterations at the expense of accuracy. To improve efficiency and accuracy of the FOM method, it is suitable to explore the defined design space and adapt convergence criteria considering features of the optimization problem.

According to the optimization problem definitions within the frame of the Subproblem Approximation Method, the dependent variable (SV) approximation techniques and effect of weighting factor were observed. The SAM method efficiency was analyzed depending on the number of random loops performed beforehand. The cases where the SV approximation was performed by quadratic and quadratic plus cross-term fitting achieve ambient of the optimum. The number of loops evaluated beforehand influence the dependent variables' function forms depending on their location within the design space. To improve stability and accuracy of the SAM method, it is recommended to explore the design space of the problem and regulate its constraints according to obtained features.

9.4 THREE-DIMENSIONAL CONSTRAINED PROBLEM

In the following, an optimization problem defined by 3 independent and 2 dependent variables is analyzed. The problem is expressed by a three-bar plane frame fixed at one end. A vertical displacement *w* of the free end [*A*] is observed with a predefined maximal permitted value. The aim of the problem is to minimize the volume of the structure by varying cross-sections' heights of all its members. The geometry and denotation of variables is pictured in Figure 9.23.

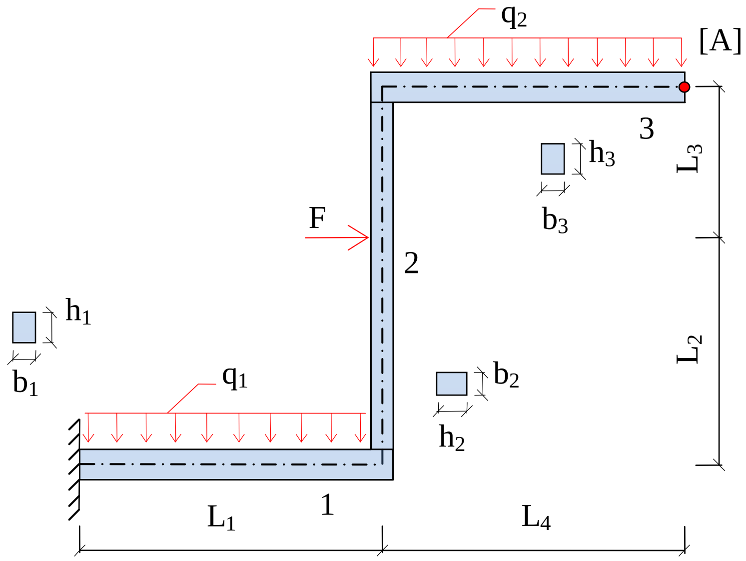

Figure 9.23 Geometry and denotation of optimization variables of problem

Geometry and varying parameters' limits, material properties of the structure and functional expressions of optimization parameters are defined as follows:

$b_1 = 0{,}2m$	$I_1 = 1{,}666\overline{6}.10^{-2}h_1^3 m^4$
$b_2 = 0{,}2m$	$I_2 = 1{,}666\overline{6}.10^{-2}h_2^3 m^4$
$b_3 = 0{,}2m$	$I_3 = 1{,}666\overline{6}.10^{-2}h_3^3 m^4$
$h_1 = \langle 0{,}1..0{,}9\rangle m$	$E = 20.10^9 Pa$
$h_2 = \langle 0{,}1..0{,}9\rangle m$	$F = 30.10^3 N$
$h_3 = \langle 0{,}1..0{,}9\rangle m$	$q_1 = 3.10^3 N/m$
$L_1 = 4m$	$q_2 = 4.10^3 N/m$
$L_2 = 3m$	$w_{[A]} = \left[\frac{11{,}2480.10^{-3}}{h_1^3} + \frac{3{,}539\overline{9}.10^{-3}}{h_2^3} + \frac{0{,}3840.10^{-3}}{h_3^3}\right] m$
$L_3 = 2m$	
$L_4 = 4m$	$f = [0{,}8h_1 + 1{,}0h_2 + 0{,}8h_3]m^3$

According to the geometry and material specification the optimization problem is defined as follows:

$$\textit{Find}\,\boldsymbol{h} = \begin{Bmatrix} h_1 \\ h_2 \\ h_3 \end{Bmatrix}, \textit{ which minimize} \quad f(\boldsymbol{h}) = 0{,}8h_1 + 1{,}0h_2 + 0{,}8h_3 \tag{9.4.1}$$

subject to:

$$\begin{gathered} 0{,}1 \le h_1 \le 0{,}9 \\ 0{,}1 \le h_2 \le 0{,}9 \\ 0{,}1 \le h_3 \le 0{,}9 \\ w_{[A]} \le 0{,}05m \end{gathered} \tag{9.4.2}$$

where $w_{[A]}$ is the vertical displacement at the point A, f is the volume of the structure and h_1, h_2 and h_3 are the cross-sections' heights of members separately. To control and verify resultant design sets obtained by the FOM and SAM method the graphical and mathematical system's expression of the optimization problem is performed.

9.4.1 Graphical Expression

A graphical expression of the optimization problem is presented in Figure 9.24 below. The optimum is situated at the adherent point of dependent variables' functions where the objective function $f(\boldsymbol{h})$ forms tangential plane of the state variable function $w_{[A]}$. The constraints of independent variables are expressed by a transparent box, which encloses the graphical expression of dependent variables.

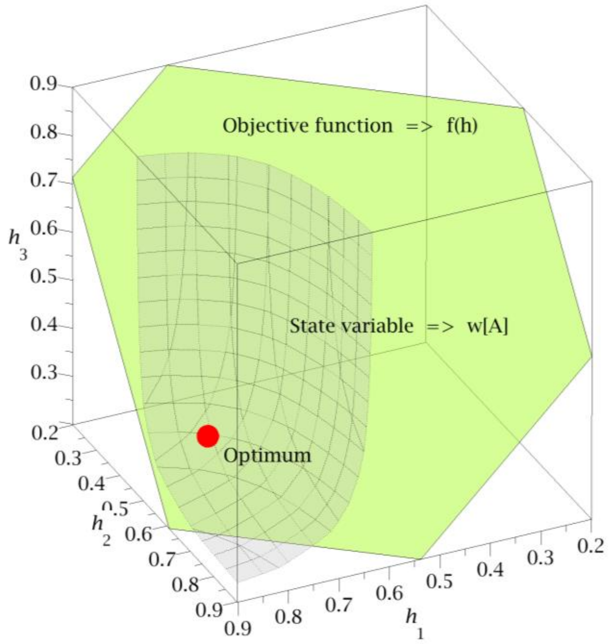

Figure 9.24 Graphical expression

9.4.2 Localization of Extreme

An analogous proceeding as in the previous optimization problem (section 9.3) was applied to localize the optimum of actual optimization problem. Then the independent optimization variables (DVs) are:

$$h_1 = 8{,}0458.\,10^{-1}\,m$$
$$h_2 = 5{,}6993.\,10^{-1}\,m$$
$$h_3 = 3{,}4585.\,10^{-1}\,m$$

and dependent variables (Obj and SV) achieve values:

$$w_{[A]} = 0{,}05\,m$$

$$f = 1{,}4903\,m^3.$$

The optimum of the presented problem is graphically expressed in Figure 9.24.

9.4.3 Solution by ANSYS/Design Optimization

According to the defined geometry (Figure 9.23) and material properties of the actual problem a finite elements' model was created. To form the FEM model two-dimensional BEAM3 elements were used. A parametrical model, which must be determined before an optimization procedure is applied, was formed. The individual parameters creating the parametrical model with combination of general optimization definition of the problem (9.4.1 and 9.4.2) are summarized in Table 9.19.

Table 9.19 Optimization variables

Variable	Expression	Description
(Obj)	f	Volume of structure
(SV)	$w_{[A]}$	Vertical displacement at point $[A]$
(DV)	h_1	Height of horizontal bar (1) cross-section
(DV)	h_2	Height of column (3) cross-section
(DV)	h_3	Height of horizontal bar (2) cross-section

In the following, the presented optimization problem is subjected to the First Order Method and Subproblem Approximation Method analyses.

9.4.4 First Order Method

A proceeding of the First Order Method (FOM) within the frame of the actual optimization problem is observed depending on different initial design set location (defined by DVs' values) and varying step lengths, which form gradients' distances determined by combination of a golden section search and a local quadratic fitting method (section 6.2.3). The initial design sets are defined at DVs' extremes and in the middle of their range. The values of optimization variables in defined initial design sets are pictured in Table 9.20.

Table 9.20 Initial design sets

Obj **[m^3]**	**SV** **[m]**	**DV$_1$** **[m]**	**DV$_2$** **[m]**	**DV$_3$** **[m]**	**Status**
0,2600	1,5940.10^{-1}	0,1	0,1	0,1	infeasible
1,3000	1,2756.10	0,5	0,5	0,5	infeasible
2,3400	2,1888.10^2	0,9	0,9	0,9	feasible

The proceeding of dependent variables, performed by the FOM method, initiated by the *DVs=0,5* due to 17 loops with different step lengths' definition is pictured in Figure 9.25 (Obj) and 9.26 (SV).

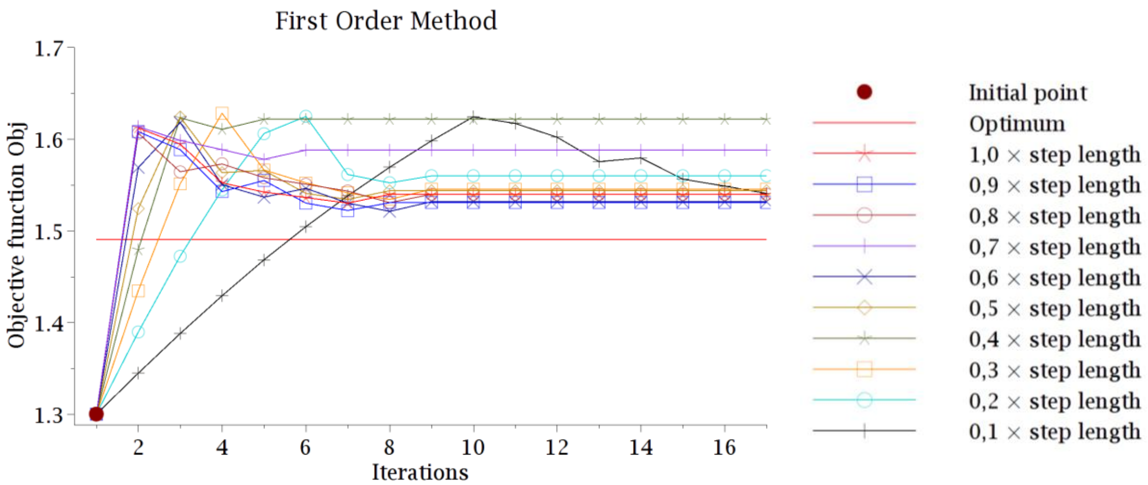

Figure 9.25 Obj proceeding by FOM

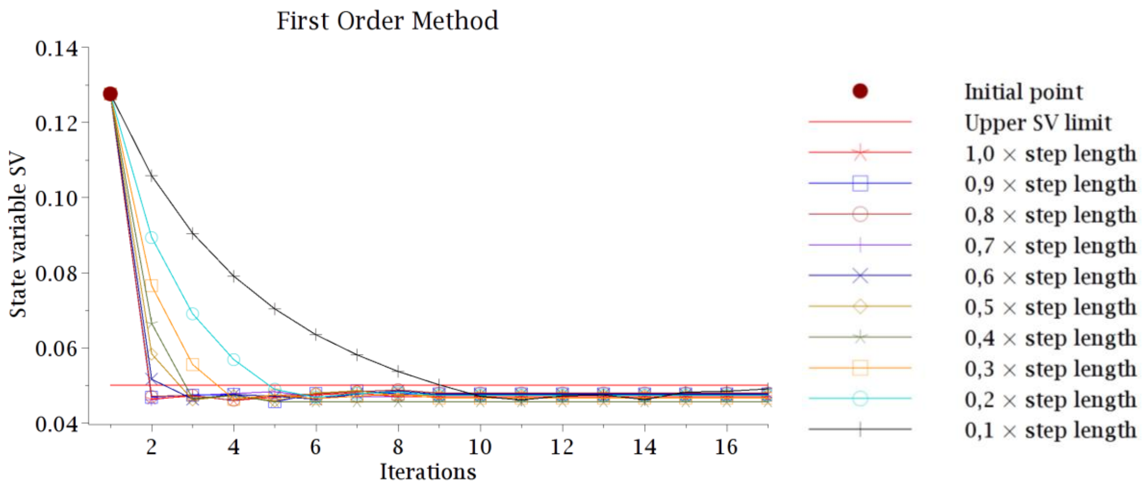

Figure 9.26 SV proceeding by FOM

The analyzed case initiated by the *DVs=0,5*, which is located in the infeasible design space, gradually tend to the optimum ambient. The features of the presented problem allow finding feasible design space after the first loop performed by the FOM method if a long step length is applied. The solution with shortest step length achieves the feasible design space after 8 loops.

The best design sets' variables' values obtained by the FOM method analysis with application of different initial points' location are summarized in Table 9.21.

Table 9.21 FOM results

	Step length [%]	**Obj [m^3]**	**SV.10^{-2} [m]**	**DV$_1$.10^{-1} [m]**	**DV$_2$.10^{-1} [m]**	**DV$_3$.10^{-1} [m]**	**Iterations [-]**
DV$_1$=0,1 DV$_2$=0,1 DV$_3$=0,1	100	1,5081	4,9689	8,2368	5,7399	3,4400	22 (93)
	90	1,5069	4,9807	8,2215	5,7160	3,4696	50 (103)
	80	1,5070	4,9813	8,2250	5,6805	3,5113	28 (85)
	70	1,5057	4,9968	8,1904	5,7908	3,3926	40 (97)
	60	1,5071	4,9790	8,2229	5,7262	3,4576	25 (121)
	50	1,5054	4,9985	8,2116	5,7696	3,3946	28 (88)
	40	1,5052	4,9975	8,1800	5,7442	3,4552	52 (78)
	30	1,5072	4,9779	8,2196	5,7226	3,4666	26 (129)
	20	1,5056	4,9931	8,2166	5,7227	3,4507	44 (129)
	10	1,5072	4,9782	8,2406	5,7139	3,4565	38 (109)
DV$_1$=0,5 DV$_2$=0,5 DV$_3$=0,5	100	1,5299	4,7928	8,5359	5,5611	3,6361	7 (13)
	90	1,5218	4,8568	8,1060	5,7754	3,6972	7 (14)
	80	1,5307	4,8853	7,7752	5,9189	3,9601	8 (14)
	70	1,5776	4,8243	7,7606	5,7020	4,8320	5 (12)
	60	1,5211	4,8569	8,2398	5,6841	3,6683	8 (15)
	50	1,5339	4,8502	7,9899	5,6893	4,0724	7 (14)
	40	1,6105	4,7381	7,5289	6,0147	5,0834	4 (11)
	30	1,5343	4,7692	8,2053	5,6849	3,8677	8 (15)
	20	1,5523	4,8240	7,8013	5,7826	4,3737	8 (15)
	10	1,5405	4,9059	7,7578	5,7735	4,2811	17 (23)
DV$_1$=0,9 DV$_2$=0,9 DV$_3$=0,9	100	1,5902	4,8489	8,0676	5,3892	5,0729	7 (13)
	90	1,5584	4,9753	8,5259	5,1295	4,5427	7 (14)
	80	1,5114	4,9402	8,2146	5,8122	3,4128	7 (13)
	70	1,5451	4,9349	7,5019	6,0908	4,1986	8 (15)
	60	1,5186	4,9286	8,0591	5,6506	3,8608	8 (16)
	50	1,7631	4,8682	7,0485	6,4519	6,9254	4 (11)
	40	1,6124	4,8360	7,8851	5,4908	5,4063	6 (13)
	30	1,7724	4,8648	7,0260	6,5064	6,9966	3 (10)
	20	1,7553	4,9363	7,0169	6,4191	6,9010	5 (12)
	10	1,6464	4,9552	7,4085	5,8032	5,9174	11 (17)

In the first case where the design variables DVs are defined $DV_1=DV_2=DV_3=0,1$ more iterations are required to achieve convergence criteria. This is caused by evaluation of gradients (section 7.2.2) due the FOM proceeding. On the other hand this solution achieves the optimum ambient with higher accuracy than the cases initiated above the optimum ($DV_1=DV_2=DV_3=0,5$ and $DV_1=DV_2=DV_3=0,9$) where the convergence criteria were acquired in markedly less loops. To improve reliability and accuracy of the solution a design space exploration by Random or Sweep optimization tool (section 7.3) is recommended.

9.4.5 Subproblem Approximation Method

Within the frame of the actual optimization problem, the Subproblem Approximation Method was analyzed depending on number of random loops, which form a set of points for the first SAM iteration, dependent variables' approximation technique and influence of different weighting factor determination. According to objective function Obj definition, which is represented by linear equation (9.4.1), its approximation technique doesn't affect the optimization process. Results from the solutions initiated by 5 random loops are summarized in Table 9.22 where superscripts assigned to the dependent variables and the weighting factor indicate appropriate settings of the SAM proceeding illustrated in Table 9.4.

Table 9.22 SAM results initiated by 5 random loops

Category	Group	Settings	Obj $[m^3]$	$SV.10^{-2}$ [m]	$DV_1.10^{-1}$ [m]	$DV_2.10^{-1}$ [m]	$DV_3.10^{-1}$ [m]	Iterations [-]
1	1	$(Obj^{0,1,2},SVs^0,W^0)$	1,5535	4,9867	8,4101	5,1766	4,5380	25 (1172)
	2	$(Obj^{0,1,2},SVs^0,W^1)$	1,5053	4,9967	8,1962	5,7204	3,4691	499 (1206)
	3	$(Obj^{0,1,2},SVs^0,W^2)$	1,5052	4,9978	8,1908	5,7134	3,4821	366 (1198)
	4	$(Obj^{0,1,2},SVs^0,W^3)$	1,5050	4,9992	8,1947	5,7114	3,4789	550 (1207)
	5	$(Obj^{0,1,2},SVs^0,W^4)$	1,5105	4,9999	8,6353	5,5040	3,3658	148 (1181)
2	6	$(Obj^{0,1,2},SVs^1,W^0)$	1,5398	4,9807	8,9719	5,9132	2,8836	39 (1174)
	7	$(Obj^{0,1,2},SVs^1,W^1)$	1,4618	6,5580	8,6264	4,4054	5,1398	230 (1164)
	8	$(Obj^{0,1,2},SVs^1,W^2)$	1,4618	6,5580	8,6264	4,4054	5,1398	230 (1164)
	9	$(Obj^{0,1,2},SVs^1,W^3)$	1,4618	6,5580	8,6264	4,4054	5,1398	230 (1164)
	10	$(Obj^{0,1,2},SVs^1,W^4)$	1,5391	4,9988	8,9699	5,9180	2,8711	43 (1177)
3	11	$(Obj^{0,1,2},SVs^3,W^0)$	1,5051	4,9990	8,1946	5,6979	3,4966	184 (1189)
	12	$(Obj^{0,1,2},SVs^3,W^1)$	1,5071	4,9831	8,1605	5,6939	3,5603	269 (1194)
	13	$(Obj^{0,1,2},SVs^3,W^2)$	1,5054	4,9971	8,2761	5,6921	3,4267	268 (1183)
	14	$(Obj^{0,1,2},SVs^3,W^3)$	1,5058	4,9999	8,2256	5,6158	3,5777	158 (1188)
	15	$(Obj^{0,1,2},SVs^3,W^4)$	1,5336	4,9947	8,9711	5,1333	3,7827	152 (1186)

The solutions where the state variable (SV) is approximated by a linear fitting lead to biased results. In any of the situations where the weighting factor tends to relative distances of design sets in the design space, or to the objective function, or else where all weighting factors are unified, the solution achieves infeasible resultant design sets (upper SV limit is exceeded). The SV approximations by the quadratic and quadratic plus cross-term curve fitting lead the solution to the ambient of the optimum despite all the random loops performed before the SAM proceeding is applied where only infeasible design sets were obtained. The following figures (Figure 9.27 and 9.28) represent dependent variables' proceeding due the first 35 iterations (5 random + 30 SAM). It can be seen in Figure 9.28, the pivotal quotient on the achieved design sets' infeasibility due the SAM proceeding, has randomly performed design set n. 3 (4^{th} in the figure because 1^{st} loop is expressed by the initial design set definition) which markedly exceeds the upper limit of the SV.

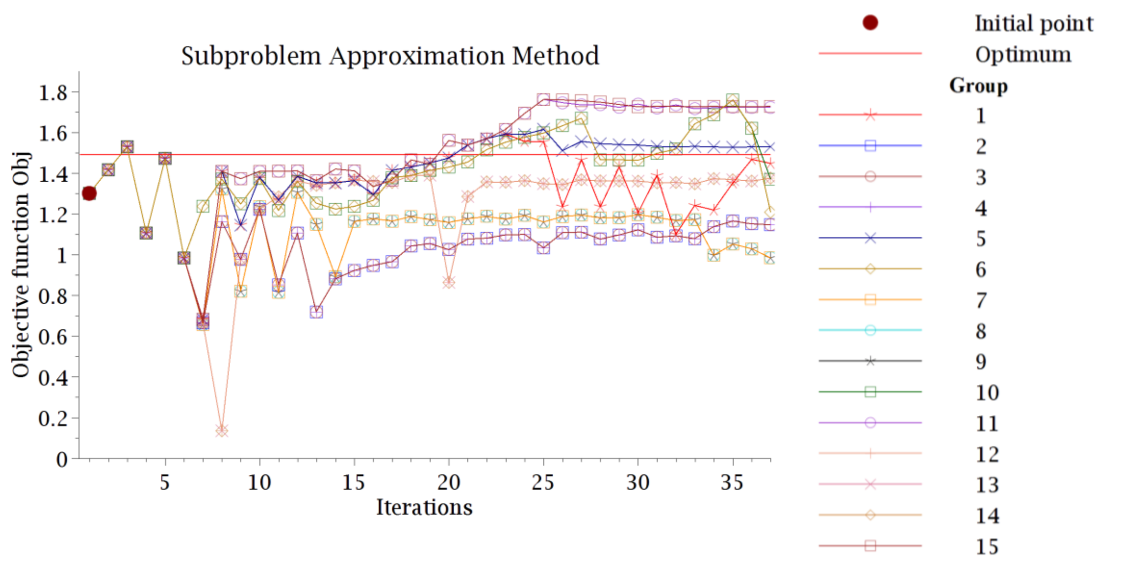

Figure 9.27 Obj proceeding by SAM

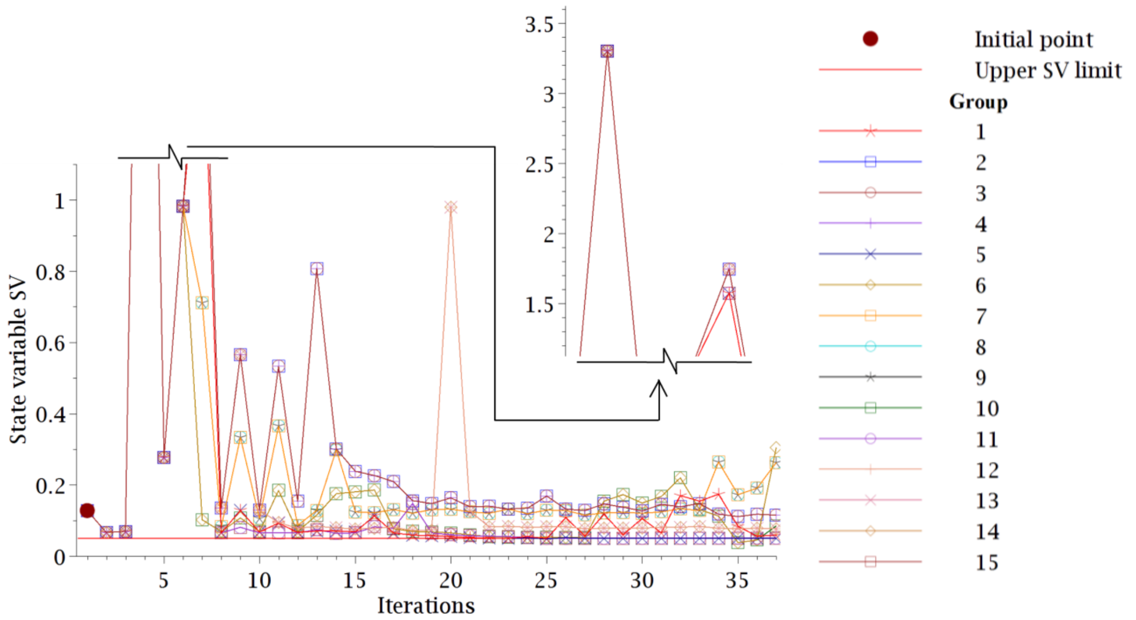

Figure 9.28 SV proceeding by SAM

The variables' values of the best design sets and number of iterations needed to achieve defined convergence by the SAM method performed initiated by 50 and 100 random loops are illustrated in the following tables (Table 9.23 and 9.24). Except best design sets in category 2 which defines solutions, where the SV is approximated by the linear curve fitting, the resultant design sets achieve an ambient of the problem optimum.

Table 9.23 SAM results initiated by 50 random loops

Category	Group	Settings	Obj $[m^3]$	SV.10^{-2} [m]	$DV_1.10^{-1}$ [m]	$DV_2.10^{-1}$ [m]	$DV_3.10^{-1}$ [m]	Iterations [-]
1	1	$(Obj^{0,1,2},SVs^0,W^0)$	1,5053	4,9962	8,2020	5,7165	3,4688	696 (1213)
	2	$(Obj^{0,1,2},SVs^0,W^1)$	1,5052	5,0000	8,1663	5,7849	3,4181	357 (1310)
	3	$(Obj^{0,1,2},SVs^0,W^2)$	1,5067	4,9989	8,0009	5,8683	3,4972	319 (1244)
	4	$(Obj^{0,1,2},SVs^0,W^3)$	1,5053	4,9965	8,1929	5,7204	3,4728	407 (1255)
	5	$(Obj^{0,1,2},SVs^0,W^4)$	1,5231	4,9791	8,9747	5,3305	3,4012	335 (1227)
2	6	$(Obj^{0,1,2},SVs^1,W^0)$	1,5706	4,6501	8,2342	6,6346	3,1051	163 (1216)
	7	$(Obj^{0,1,2},SVs^1,W^1)$	1,5513	4,9305	8,9692	6,0643	2,8410	247 (1247)
	8	$(Obj^{0,1,2},SVs^1,W^2)$	1,5132	4,9911	8,7089	5,5298	3,2935	189 (1250)
	9	$(Obj^{0,1,2},SVs^1,W^3)$	1,5759	4,9099	8,9697	6,4004	2,7279	147 (1211)
	10	$(Obj^{0,1,2},SVs^1,W^4)$	1,6377	4,8633	7,0691	6,7725	4,9360	53 (1211)
3	11	$(Obj^{0,1,2},SVs^3,W^0)$	1,5053	4,9969	8,2122	5,7106	3,4652	243 (1220)
	12	$(Obj^{0,1,2},SVs^3,W^1)$	1,5052	4,9993	8,1716	5,7738	3,4264	374 (1305)
	13	$(Obj^{0,1,2},SVs^3,W^2)$	1,5055	4,9985	8,0928	5,7640	3,5213	211 (1469)
	14	$(Obj^{0,1,2},SVs^3,W^3)$	1,5050	4,9996	8,1883	5,7076	3,4898	299 (1230)
	15	$(Obj^{0,1,2},SVs^3,W^4)$	1,5058	4,9999	8,3376	5,6014	3,4828	274 (1229)

The beginning of the SAM processing is influenced by the random design sets' locations. The following figure (Figure 9.29) illustrates the SV values proceeding due the procedure performed by the Random Tool. Among the first 50 random loops only 4 achieve feasible conditions and in the case of 100 randomly performed loops it is 7. It can be seen that the defined DVs' constraints allows to the Random Tool create design sets which markedly exceed the upper SV limit (9.4.2). These then significantly influence initial proceeding performed by the SAM method.

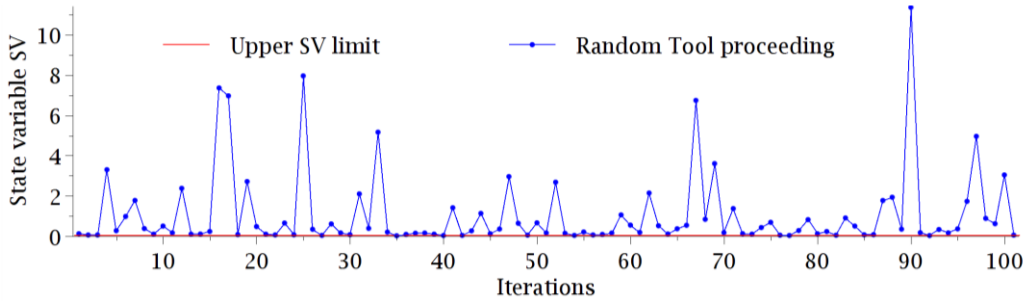

Figure 9.29 SV proceeding by SAM

The SAM method proceeding initiated by 100 random loops is analogical to the previous case where 50 random loops were performed before the SAM procedure was applied because of features' similarity in both initial processes.

Table 9.24 SAM results initiated by 100 random loops

Category	Group	Settings	Obj [m^3]	SV.10^{-2} [m]	$DV_1.10^{-1}$ [m]	$DV_2.10^{-1}$ [m]	$DV_3.10^{-1}$ [m]	Iterations [-]
1	1	$(Obj^{0,1,2},SVs^0,W^0)$	1,5052	4,9976	8,1961	5,7176	3,4716	1203 (1273)
	2	$(Obj^{0,1,2},SVs^0,W^1)$	1,5056	4,9972	8,2944	5,6466	3,4668	374 (1435)
	3	$(Obj^{0,1,2},SVs^0,W^2)$	1,5051	4,9980	8,2060	5,7090	3,4720	738 (1395)
	4	$(Obj^{0,1,2},SVs^0,W^3)$	1,5052	4,9978	8,2122	5,7112	3,4634	542 (1309)
	5	$(Obj^{0,1,2},SVs^0,W^4)$	1,5119	4,9926	8,6094	5,4282	3,5037	139 (1321)
2	6	$(Obj^{0,1,2},SVs^1,W^0)$	1,5212	4,9980	8,9638	5,5012	3,1751	217 (1434)
	7	$(Obj^{0,1,2},SVs^1,W^1)$	1,5296	4,9264	8,9614	5,6437	3,1041	250 (1285)
	8	$(Obj^{0,1,2},SVs^1,W^2)$	1,5288	4,8943	8,9616	5,4692	3,3117	130 (1334)
	9	$(Obj^{0,1,2},SVs^1,W^3)$	1,5245	4,9737	8,9699	5,2950	3,4682	152 (1282)
	10	$(Obj^{0,1,2},SVs^1,W^4)$	1,5264	4,9693	8,8237	5,7862	3,0234	113 (1270)
3	11	$(Obj^{0,1,2},SVs^3,W^0)$	1,5051	4,9980	8,2091	5,7189	3,4567	303 (1282)
	12	$(Obj^{0,1,2},SVs^3,W^1)$	1,5057	4,9997	8,3474	5,6530	3,4167	570 (1419)
	13	$(Obj^{0,1,2},SVs^3,W^2)$	1,5051	5,0000	8,2624	5,7100	3,4144	330 (1304)
	14	$(Obj^{0,1,2},SVs^3,W^3)$	1,5052	4,9973	8,2011	5,7178	3,4667	368 (1288)
	15	$(Obj^{0,1,2},SVs^3,W^4)$	1,5220	4,9988	8,9748	5,5178	3,1524	162 (1312)

The great number of iterations needed to achieve convergence is caused by large definition of design space (DVs' constraints) and location of design sets obtained by the Random Tool before the SAM method is applied. Within the frame of the presented optimization problem more random iterations don't increase efficiency of the SAM method.

9.4.6 Discussion and Conclusion

An efficiency of the FOM and SAM methods was analyzed within the frame of strictly convex optimization problem expressed by a three-bar plane frame fixed at one end. The aim was finding minimum volume of the structure constrained by lower and upper limits of DVs (cross sections' characteristics) and upper limit of SV (structure displacement of the free end).

The First Order Method was analyzed depending on different initial design set location, and the optimization processing was controlled by varying gradients' step lengths. In the case that the initial point is located in the infeasible design space under actual optimum point the gradients' formation causes the solution to require markedly more iterations to achieve convergence criteria. On the other hand the resultant design sets acquire more accurate values compared to the solution where the initial points are located above the minimum. The efficiency of step length determination depends on certain optimization problem definition and the initial point location. To improve the efficiency of the optimization procedure performed by the FOM, design space exploration in advance is recommended.

The optimization processing performed by the Subproblem Approximation Method was observed depending on different approximation technique of the state variable (SV), weighting factor definition and number of random loops performed in advance. The accuracy of achieved best design sets and number of iterations needed to achieve convergence criteria is mostly influenced by weighting factor definition. The state variable linear fitting approximation causes less accurate results in comparison with quadratic or quadratic plus cross-term approximation. Within the frame of the presented problem the number of loops performed by the Random Tool, before the SAM is proceeding doesn't significantly influence obtained resultant design sets.

9.6 EFFICIENT DESIGN OF TRUSS BEAM

The aim of the following problem is to design a truss beam with minimal weight according to the geometry presented in Figure 9.30. The structure is loaded by vertical forces in upper joints and by its own weight. It is supported on the bottom edges' joints so that the left-most joint is avoided in horizontal and vertical displacements and the right-most joint is avoided in vertical displacement only. According to the problem specification the members 1, 4 and 6 are subjected to tensile strain and members 2, 3 and 5 to compressive strain. Design conditions of the problem are stated by EN 1993 Eurocode 3 [78], for uniform compression forces in compression members and tensile resistance moments in tension members. Furthermore, a minimal frequency of the first natural modal shape of the structure is defined.

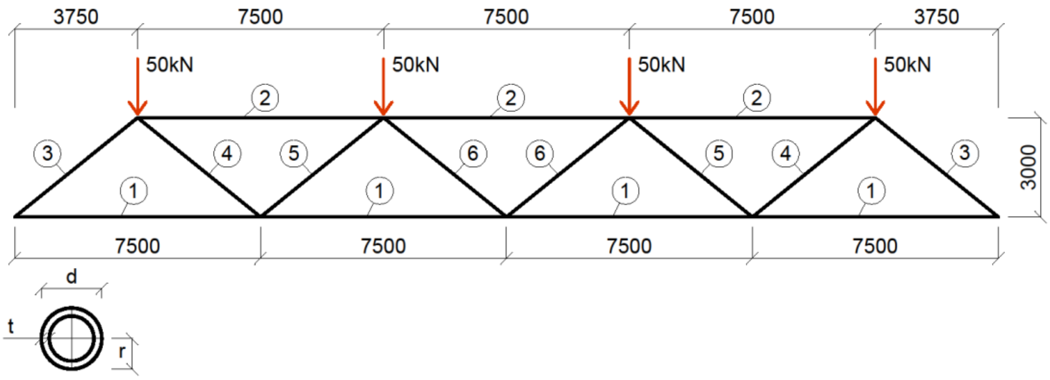

Figure 9.30 Truss Beam

The bottom and upper chords' cross-sections are constant along the length of the structure. With regard to the symmetry of the beam the members are paired according to their loading. Then the design consists of designing 6 members of the truss-beam (Figure 9.30). The minimal weight of the structure is achieved by minimizing of truss-beam members' cross-sectional areas. All members of the structure are of tubular cross-sections with fixed wall thickness (4mm).

9.6.1 Problem Definition

According to defined conditions and restrictions, an optimization problem definition is formed. Independent variables are defined by cross-sectional properties of the truss members. The members have to satisfy resistances which are determined by Eurocode 3. Furthermore, a frequency of first natural mode shape is limited. The objective function is expressed by a volume of the truss beam.

The optimization problem is defined as follows:

$$Find\,\boldsymbol{r} = \begin{Bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{Bmatrix}, which\ minimize\ f(\boldsymbol{r}) \tag{9.5.1}$$

subject to:

$$\begin{array}{lll} \underline{r}_1 \le r_1 \le \overline{r}_1 & N_{t,Rd}{}^{(1)} \ge N_{t,Ed}{}^{(1)} & N_{c,Rd}{}^{(2)} \ge N_{Ed}{}^{(2)} \\ \underline{r}_2 \le r_2 \le \overline{r}_2 & N_{t,Rd}{}^{(4)} \ge N_{t,Ed}{}^{(4)} & N_{c,Rd}{}^{(3)} \ge N_{Ed}{}^{(3)} \\ \underline{r}_3 \le r_3 \le \overline{r}_3 & N_{t,Rd}{}^{(6)} \ge N_{t,Ed}{}^{(6)} & N_{c,Rd}{}^{(5)} \ge N_{Ed}{}^{(5)} \\ \underline{r}_4 \le r_4 \le \overline{r}_4 & & \\ \underline{r}_5 \le r_5 \le \overline{r}_5 & Modal \ge 3Hz & \\ \underline{r}_6 \le r_6 \le \overline{r}_6 & & \end{array} \tag{9.5.2}$$

where r_i are internal radiuses of members' cross-sections, $\underline{r}_i$ and $\overline{r}_i$ are design variables' constraints, i number of design variables, N_{Ed} is the design value of the compression force and $N_{t,Ed}$ is the design tensile force.

The tensile resistances of members 1, 4 and 6 (Figure 9.30) correspond to:

$$N_{t,Rd} = \frac{f_y}{\gamma_{M1}} A \tag{9.5.3}$$

where f_y *=235N/mm²*, γ_{M1}*=1,15* and A is a net cross-sectional area of the members.

The dependent variables, which represent compression members' resistance, are defined:

$$N_{c,Rd} = \chi \frac{A.f_y}{\gamma_{M1}} \tag{9.5.4}$$

The reduction factor corresponds to:

$$\chi = \frac{1}{\phi + \left(\phi^2 - \bar{\lambda}^2\right)^{0,5}} \tag{9.5.5}$$

where

$$\phi = 0,5\left(1 + \alpha\left(\bar{\lambda} - 0,2\right) + \bar{\lambda}^2\right) \tag{9.5.6}$$

and non-dimensional slenderness $\bar{\lambda}$ is expressed by:

$$\bar{\lambda} = \frac{L_{cr}}{i}\frac{1}{\lambda_1} \tag{9.5.7}$$

$$\lambda_1 = \pi \sqrt{\frac{E}{f_y}} \tag{9.5.8}$$

9.6.2 FEM Model and Optimization Variables Definition

The analyzed optimization methods are iterative procedures. Each optimization loop requires the entire structure of one complex FEM/FEA analysis. For this reason an analysis file (section 8.4.2) is created within the frame of the presented problem. The analysis file includes the definition of a parametrical model, solution process, results obtained and parameterization of dependent quantities. The parametrical model of the truss beam is expressed by the FEM/FEA model assembled from the BEAM 44 elements which are characterized by six degrees of freedom at each node. The dimensions of the problem are defined by members' axial lengths. Members' cross-sections are determined by tubular sectional type in the preprocessor (PREP 7) whose characteristics represent independent variables (design variables DVs) in the optimization process. The truss beam is simply supported at the bottom edge joints. At the top joints of the truss beam on-plane displacements, which simulate purlins' connections, are avoided. Loads carried by the purlins are defined by single vertical forces applied at the top nodes of the structure. Within the frame of the actual analysis the structure's own weight is obtained by applying a vertical linear acceleration. According to the defined conditions (9.5.2) of the design, two solutions must be performed in terms of each optimization loop. At first the parametrical model (with no external loading applied) is subjected to a modal analysis by the Subspace iteration method to obtain the first modal shape frequency of the problem. The second solution is realized by static (steady state) analysis with external loading forces applied. Both solutions are evaluated in postprocessor (POST 1) separately. Where needed, dependent quantities are parameterized. The independent (DVs) and dependent (SVs and Obj) optimization variables which are within the frame of the analysis file of the current problem defined are summarized in Table 9.25.

Table 9.25 Optimization variables

Variable	Expression	Description
(Obj)	f	Volume of structure
(SV)	N_1	max strain in bottom chords 1
(SV)	N_4	max strain in upper chords 2
(SV)	N_6	max strain in members 3
(SV)	P_2	max strain in members 4
(SV)	P_3	max strain in members 5
(SV)	P_5	max strain in members 6
(SV)	$Modal$	min 1st natural frequency
(DV)	r_1	Int. radius of bottom chord (1)
(DV)	r_2	Int. radius of upper chord (2)
(DV)	r_3	Int. radius of members (3)
(DV)	r_4	Int. radius of members (4)
(DV)	r_5	Int. radius of members (5)
(DV)	r_6	Int. radius of members (6)

9.6.3 First Order Method

At first the truss beam design is performed by applying the First Order Method (FOM) (section 7.2). The constraints of the dependent variables are determined by Eurocode 3 (eqs. 9.5.3 to 9.5.8) and fixed minimum value of the first natural shape frequency (9.5.2). Within the frame of rough exploration of the optimization problem's design space performed by the Sweep and Factorial tool (section 7.3) the design variables' (DVs) constraints were defined as follows:

$$\begin{array}{ll} 0{,}005m \leq r_1 \leq 0{,}2m & 0{,}00401m \leq r_4 \leq 0{,}15m \\ 0{,}005m \leq r_2 \leq 0{,}18m & 0{,}00401m \leq r_5 \leq 0{,}15m \\ 0{,}00401m \leq r_3 \leq 0{,}19m & 0{,}00401m \leq r_6 \leq 0{,}15m \end{array} \qquad (9.5.9)$$

The truss beam design was analyzed considering three different initial design sets (expressed by DVs' values) and varying step lengths which define gradients' distances (section 6.2.3) in the FOM iterative procedure. The initial design sets are located at lower and upper DVs' constraints and in the middle of their ranges. The step lengths are graded in *10%* of maximal size (section 7.2.2) which is determined by a combination of golden section search and local quadratic fitting method (section 6.2.3). The initial design sets defined on the lower constraints of the DVs and in the middle of their range are located in the infeasible design space. This leads to a large value of the internally evaluated objective function value, which is caused by the large value of penalty function assigned to one or more state variables. Therefore, the aim at the beginning of these solutions is to decrease penalty functions' values (section 7.2.1) until the feasible design sets are achieved.

The progressions of the FOM solution initiated by different design sets considering various step lengths are pictured in Figure 9.31.

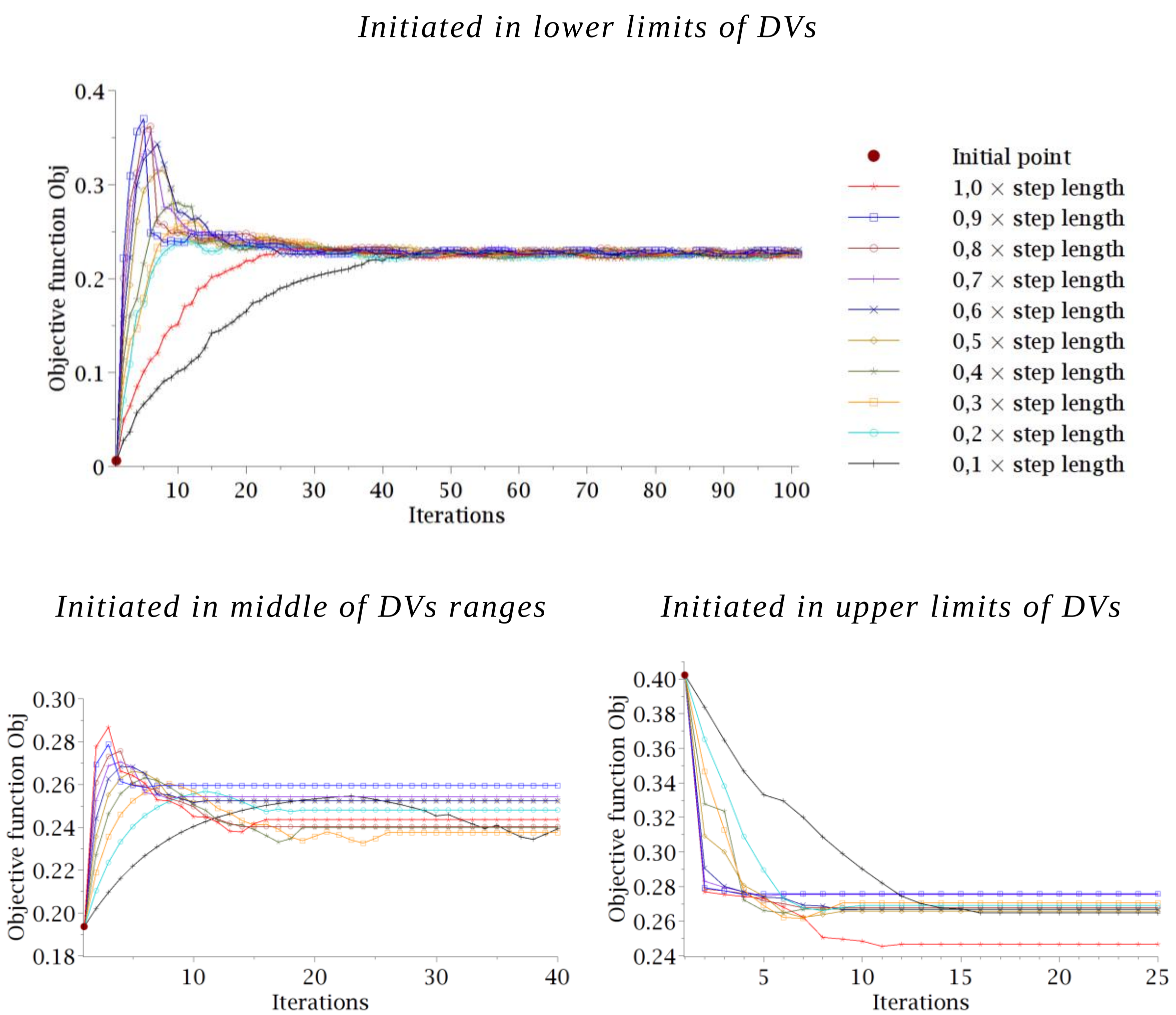

Figure 9.31 FOM proceeding

In the case that the solution is initiated under optimal objective function level the FOM method requires markedly more iterations to achieve convergence. This is caused by gradients' values' formation that must be evaluated to iterate towards the optimum. The Obj and DVs' values achieved in the best design sets within the frame of each performed solution and numbers of optimization loops performed by the FOM method are summarized in Table 9.26.

Table 9.26 FOM best design sets in rough explored DVs' intervals

Step length [%]	**Obj.10^{-1} [m^3]**	**$DV_1.10^{-2}$ [m]**	**$DV_2.10^{-1}$ [m]**	**$DV_3.10^{-1}$ [m]**	**$DV_4.10^{-2}$ [m]**	**$DV_5.10^{-2}$ [m]**	**$DV_6.10^{-2}$ [m]**	**Iterations [-]**
Init	*0,0590*	*0,5000*	*0,0500*	*0,0401*	*0,4010*	*0,4010*	*0,4010*	
100	2,2823	5,6575	1,6537	1,7491	8,0029	5,9968	8,6129	31 (112)
90	2,2851	5,6503	1,6589	1,7469	8,1417	5,6885	8,8215	100 (129)
80	2,2847	5,6371	1,6578	1,7594	7,9850	5,8131	8,7761	42 (129)
70	2,2854	5,6456	1,6712	1,7398	7,8004	5,9413	8,7176	50 (129)
60	2,2856	5,6551	1,6642	1,7396	8,0875	5,7518	8,7659	101 (112)
50	2,2483	5,6829	1,7107	1,7877	2,0742	8,6567	8,6708	87 (129)
40	2,2737	5,6524	1,6990	1,7433	9,1425	2,6733	9,4518	48 (129)
30	2,2494	5,6587	1,6943	1,7595	2,0864	8,7383	9,3624	36 (129)
20	2,2461	5,6879	1,6983	1,7988	2,0712	8,7046	8,6991	69 (129)
10	2,2472	5,6843	1,6963	1,7963	2,0745	8,7757	8,7534	101 (129)
Init	*1,9358*	*9,7500*	*0,8750*	*0,9300*	*7,2995*	*7,2995*	*7,2995*	
100	2,4230	6,0184	1,6040	1,8005	7,9877	9,9952	9,9858	12 (26)
90	2,5857	6,4823	1,7075	1,8462	8,8247	10,7470	10,8100	6 (17)
80	2,4177	6,0632	1,6292	1,7550	8,1076	9,6187	9,7413	13 (23)
70	2,5341	6,3781	1,7114	1,8602	8,5615	9,8448	9,9334	8 (19)
60	2,5146	6,365	1,7302	1,8373	8,4301	9,3423	9,5868	10 (20)
50	2,5180	6,3784	1,7381	1,8126	8,3805	9,4910	9,6477	10 (21)
40	2,3887	6,0215	1,6347	1,7779	7,9664	8,9333	9,1461	15 (29)
30	2,3642	6,0646	1,6433	1,7483	7,7803	8,3798	8,8291	22 (36)
20	2,4731	6,3612	1,7007	1,7878	8,3486	9,1153	9,3746	18 (29)
10	2,3801	6,0011	1,6612	1,7185	8,0500	8,6786	8,9936	36 (50)
Init	*4,0219*	*20,000*	*1,8000*	*1,9000*	*15,000*	*15,000*	*15,000*	
100	2,4541	6,3958	1,5634	1,7803	1,1095	9,5340	8,604	11 (18)
90	2,7530	6,7040	1,5873	1,8701	1,2880	13,6100	12,715	4 (15)
80	2,6764	6,4952	1,5422	1,8429	1,2613	13,1490	12,246	7 (17)
70	2,7462	6,6739	1,5836	1,8662	1,2875	13,5730	12,691	5 (15)
60	2,6645	6,4927	1,5187	1,8218	1,2697	13,2040	12,383	9 (16)
50	2,6371	6,4517	1,5093	1,7764	1,2701	13,0010	12,249	8 (19)
40	2,6614	6,3825	1,5112	1,7856	1,2986	13,3430	12,711	5 (17)
30	2,6912	6,4221	1,5317	1,8059	1,3194	13,5280	12,744	5 (19)
20	2,6772	6,4016	1,5267	1,8009	1,3104	13,2940	12,721	7 (20)
10	2,6471	6,4295	1,4995	1,7712	1,3083	13,0810	12,554	16 (25)

In the case that the initial design set is defined in the infeasible design space (generally under optimum objective function level), the FOM method achieves better designs of the problem than in the case where the initial point is defined in the feasible design space.

If more detailed exploration of the problem is performed by optimization tool(s) (section 7.3) before the FOM method is applied, the DVs' constraints create smaller design space. The exploration performed within the frame of the actual problem (Sweep, Random and Factorial Tool) leads to the following DVs' constraints:

$$\begin{aligned} &0{,}03m \le r_1 \le 0{,}08m && 0{,}02m \le r_4 \le 0{,}15m \\ &0{,}12m \le r_2 \le 0{,}18m && 0{,}02m \le r_5 \le 0{,}15m \\ &0{,}16m \le r_3 \le 0{,}20m && 0{,}05m \le r_6 \le 0{,}15m \end{aligned} \tag{9.5.10}$$

Then the optimization problem's (9.5.1) solution subjected to constraints (9.5.10) is analogically performed. Initial design sets are selected on the lower and upper boundaries of the DVs and in the middle of their ranges. Progresses of all analyzed solutions are pictured in Figure 9.32.

If narrower constraints of DVs are applied the differences of required iterations to achieve convergence criteria of solutions initiated in different locations of the design space aren't so appreciable against the previous one. Progressions of the particular solutions lead to the feasible design space approximately to an assumed optimum level. The largest deviation is achieved in the case where the initial point is located in the middle of DVs' ranges and step length is defined by *80%* of the maximal length.

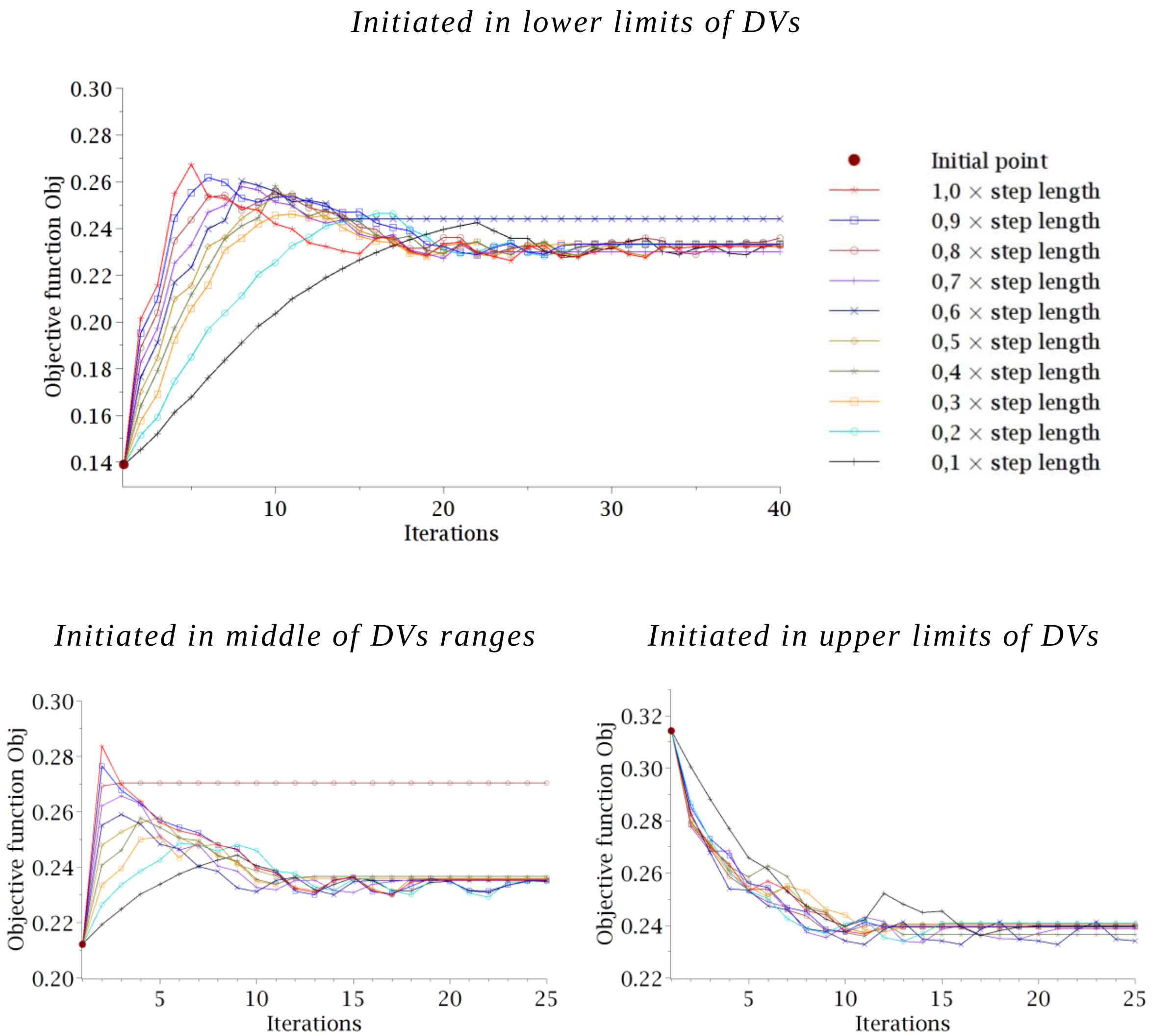

Figure 9.32 FOM proceeding in more precisely explored design space

The summarization of Obj and DVs' values obtained in the solution of narrower design space is pictured in the following table (Table 9.27).

Table 9.27 FOM best design sets in narrower DVs' intervals

Step length [%]	**Obj.10^{-1} [m^3]**	**$DV_1.10^{-2}$ [m]**	**$DV_2.10^{-1}$ [m]**	**$DV_3.10^{-1}$ [m]**	**$DV_4.10^{-2}$ [m]**	**$DV_5.10^{-2}$ [m]**	**$DV_6.10^{-2}$ [m]**	**Iterations [-]**
Init	*1,3891*	*3,0000*	*1,2000*	*1,6000*	*2,0000*	*2,0000*	*2,0000*	
100	2,3139	5,9337	1,7071	1,7441	9,2953	2,8384	9,7267	34 (44)
90	2,3176	5,8931	1,6834	1,7525	8,2245	5,0603	9,3260	23 (37)
80	2,3131	5,9006	1,6815	1,7449	7,8605	5,6102	9,0484	24 (68)
70	2,3062	5,8444	1,6804	1,7488	8,0968	5,3041	8,9951	18 (33)
60	2,4326	6,2655	1,7048	1,6509	7,5749	8,7880	10,3670	7 (24)
50	2,3139	5,8886	1,6823	1,7168	7,9025	5,9891	8,9624	29 (44)
40	2,3246	5,9092	1,6813	1,7510	8,1171	5,7819	9,0159	25 (40)
30	2,3261	5,9235	1,7018	1,6843	7,0538	7,1102	8,9532	26 (35)
20	2,3258	5,9264	1,6718	1,7026	8,0128	6,6272	8,9786	23 (39)
10	2,3239	5,9602	1,6902	1,6907	7,7437	6,5491	8,8297	36 (50)
Init	*2,1208*	*5,5000*	*1,5000*	*1,8000*	*6,5000*	*6,5000*	*6,5000*	
100	2,3512	6,1036	1,6578	1,8303	8,1289	5,9451	9,0924	14 (29)
90	2,3346	6,0559	1,6552	1,8233	7,9528	5,7259	9,0787	23 (34)
80	2,6931	7,5000	1,6999	1,8217	10,6360	10,4620	10,9830	2 (14)
70	2,3483	6,0825	1,6819	1,8161	7,8445	6,0112	8,8362	17 (27)
60	2,3514	6,1129	1,6508	1,8140	8,0501	6,6449	8,7808	11 (26)
50	2,3508	6,1018	1,6638	1,8394	8,3106	5,7961	8,8164	14 (29)
40	2,3550	6,0563	1,6665	1,8237	8,2331	6,5404	8,5586	10 (23)
30	2,3479	6,0453	1,6424	1,8150	8,2043	7,1034	8,4187	10 (22)
20	2,3473	6,0838	1,6730	1,8110	7,8565	6,1132	8,9327	24 (33)
10	2,3442	6,0947	1,6511	1,8125	8,0209	6,6401	8,5801	19 (34)
Init	*3,1420*	*8,0000*	*1,8000*	*2,0000*	*15,0000*	*15,0000*	*15,0000*	
100	2,3779	6,2810	1,6831	1,9454	8,1352	4,9342	8,9049	10 (23)
90	2,3854	6,2837	1,6395	1,9470	8,2896	5,6479	9,3483	9 (22)
80	2,3805	6,2999	1,6637	1,9441	8,1039	5,2665	9,1242	9 (22)
70	2,3705	6,2856	1,6456	1,9182	7,8195	5,6666	9,3185	20 (30)
60	2,3770	6,3397	1,6457	1,9476	8,1872	5,2871	9,1372	9 (39)
50	2,3856	6,3390	1,6490	1,9425	8,0916	5,5518	9,2982	12 (23)
40	2,3729	6,2556	1,6440	1,9370	8,0373	5,5282	9,2771	10 (24)
30	2,3848	6,3048	1,6705	1,9457	8,0163	5,3084	9,1549	11 (24)
20	2,4057	6,3000	1,5839	1,9364	8,4335	6,9301	10,1210	10 (25)
10	2,3926	6,2801	1,5841	1,9346	8,2880	6,7408	9,9859	19 (28)

The FOM method applied in the truss beam design points out that a more detailed exploration of the design space with accompanying of narrower DVs' constraints doesn't guarantee more efficient design. Within the frame of performed analysis the best solutions are achieved in the cases where the design space is briefly explored, initial design set is defined by lower boundaries of DVs' constraints and *10-50%* of maximal gradients' lengths are applied.

9.6.4 Subproblem Approximation Method

In the following, the truss-beam design is performed by the Subproblem Approximation Method (section 7.1). Different features of the method are applied to achieve efficient design of the structure. The SAM method is based on approximation of dependent variables' functions (Obj and SVs) and uses weight factors' utilization in advantage to reach more efficient progress of the optimization procedure. The analysis was performed according to combination of the available method's features pictured in the Table 9.4 (section 9.1.9). The aim of the presented problem is minimizing truss-beam volume (weight) subjected to defined conditions (9.5.2). The truss-beam volume represents objective function (9.5.1) of the problem and it is expressed by a linear formula. This leads to changeless linear form of the objective function (Obj) in each loop performed by the SAM method for different approximations' features (Table 9.4). At first the optimization process of the SAM method is initiated by 50 and 100 loops performed by the Random Tool (section 7.3.1) and constraints of the problem correspond to conditions expressed by inequations (9.5.9) above. The proceedings for cases where the solution is initiated by 100 random loops, SVs' functions are approximated by quadratic plus cross-term curve fitting and different weights are applied, are pictured in Figure 9.33.

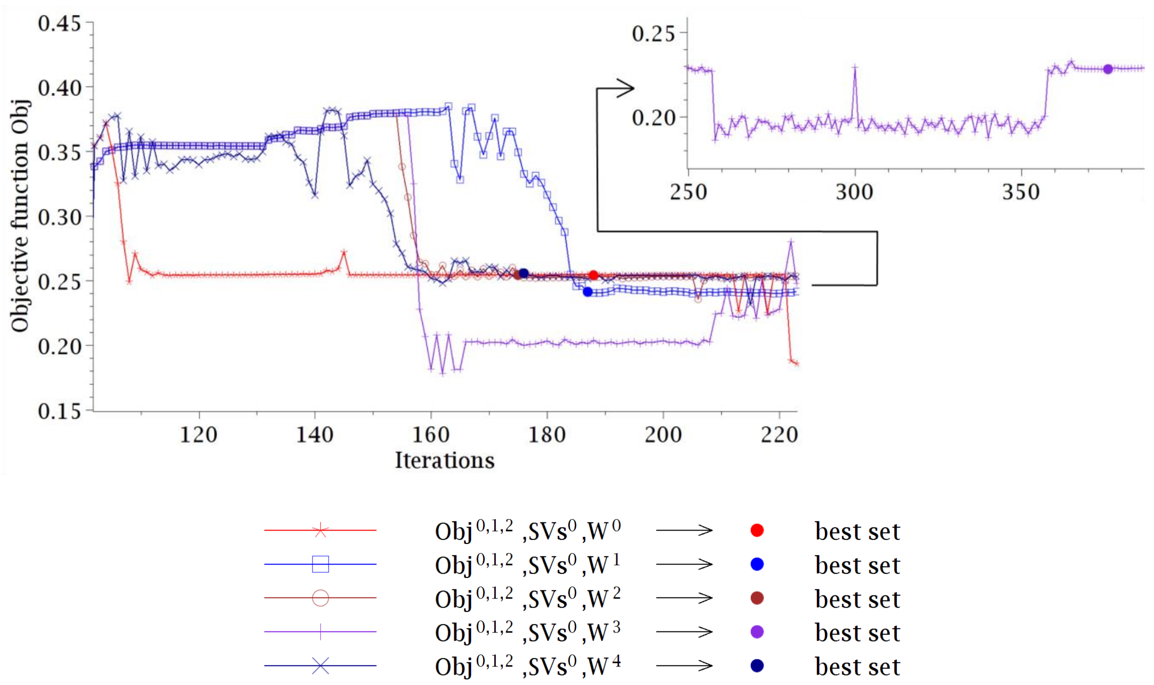

Figure 9.33 SAM proceeding initiated by 100 random loops

The weights applied to different terms of the problem (Table 9.4) considerably influence the SAM procedure of searching minimal objective function. Nevertheless, the best design sets obtained in each of the solution achieve reciprocal results. The most efficient results are obtained in the solution where the weight factor is directed to objective function values at the expense of the number of iterations which are needed to achieve convergence criteria.

The objective function (Obj) and design variables' (DVs) values of the best design sets obtained in the solution performed by the SAM method considering constraints (9.5.9) are summarized in the following tables (Table 9.29 and 9.30), where Table 9.28 represents results of the solution initiated by 50 random loops and Table 9.29 by 100 random loops. The solutions mutually deviate from an approximation method of SVs (linear, quadratic or quadratic plus cross-term curve fitting) and different approach of the weight factor.

Table 9.28 SAM best design sets in rough explored DVs' int. (init. by 50 random loops)

Settings	$Obj.10^{-1}$ $[m^3]$	$DV_1.10^{-2}$ [m]	$DV_2.10^{-1}$ [m]	$DV_3.10^{-1}$ [m]	$DV_4.10^{-2}$ [m]	$DV_5.10^{-2}$ [m]	$DV_6.10^{-2}$ [m]	Iterations [-]
$(Obj^{0,1,2},SVs^0,W^0)$	2,5672	6,2777	1,5598	1,8959	9,0595	14,9250	9,2338	181 (467)
$(Obj^{0,1,2},SVs^0,W^1)$	2,4319	5,9207	1,4607	1,8954	13,5350	9,5355	7,9814	175 (466)
$(Obj^{0,1,2},SVs^0,W^2)$	2,4290	5,9254	1,4661	1,8899	13,7790	9,2264	7,8398	179 (467)
$(Obj^{0,1,2},SVs^0,W^3)$	2,3669	5,9208	1,5328	1,8500	10,7040	8,3099	8,1072	302 (510)
$(Obj^{0,1,2},SVs^0,W^4)$	2,4235	5,9295	1,6571	1,8958	2,1381	14,9690	8,9611	177 (480)
$(Obj^{0,1,2},SVs^1,W^0)$	2,3988	5,9279	1,5965	1,8962	8,8136	2,6841	14,9690	96 (546)
$(Obj^{0,1,2},SVs^1,W^1)$	2,5224	6,3880	1,6604	1,8961	8,5647	5,1295	14,9640	177 (474)
$(Obj^{0,1,2},SVs^1,W^2)$	2,4967	5,9210	1,5607	1,8936	13,7960	2,6508	14,9680	183 (490)
$(Obj^{0,1,2},SVs^1,W^3)$	2,4729	6,3356	1,6493	1,8956	4,6155	7,4882	14,9330	423 (567)
$(Obj^{0,1,2},SVs^1,W^4)$	3,6446	19,116	1,7932	1,8963	14,9190	2,6276	14,7680	59 (459)
$(Obj^{0,1,2},SVs^3,W^0)$	2,4700	5,9208	1,6337	1,8957	12,2690	11,1330	5,1729	159 (477)
$(Obj^{0,1,2},SVs^3,W^1)$	2,5201	5,9209	1,4417	1,8760	14,7720	14,3570	6,2202	196 (489)
$(Obj^{0,1,2},SVs^3,W^2)$	2,5328	5,9211	1,4070	1,8940	14,9640	14,9670	6,5722	166 (468)
$(Obj^{0,1,2},SVs^3,W^3)$	2,4608	5,9278	1,6322	1,8961	14,2700	8,3789	5,5520	163 (479)
$(Obj^{0,1,2},SVs^3,W^4)$	2,4866	5,9205	1,5883	1,8959	14,9690	10,1110	5,2446	171 (478)

Table 9.29 SAM best design sets in rough explored DVs' int. (init. by 100 random loops)

Settings	$Obj.10^{-1}$ $[m^3]$	$DV_1.10^{-2}$ [m]	$DV_2.10^{-1}$ [m]	$DV_3.10^{-1}$ [m]	$DV_4.10^{-2}$ [m]	$DV_5.10^{-2}$ [m]	$DV_6.10^{-2}$ [m]	Iterations [-]
$(Obj^{0,1,2},SVs^0,W^0)$	2,5405	5,9207	1,4160	1,8948	13,2780	14,9710	8,3572	188 (526)
$(Obj^{0,1,2},SVs^0,W^1)$	2,4122	5,9367	1,6472	1,8886	2,0717	14,9510	8,8605	187 (535)
$(Obj^{0,1,2},SVs^0,W^2)$	2,5423	5,9207	1,3794	1,8961	14,9700	14,9690	7,5850	175 (522)
$(Obj^{0,1,2},SVs^0,W^3)$	2,2808	5,9208	1,6661	1,8746	2,0714	9,7204	8,3926	376 (531)
$(Obj^{0,1,2},SVs^0,W^4)$	2,5553	5,9209	1,3764	1,8963	14,9710	14,9700	8,1905	176 (534)
$(Obj^{0,1,2},SVs^1,W^0)$	2,5460	5,9211	1,3558	1,8944	14,9630	14,9640	8,3226	222 (544)
$(Obj^{0,1,2},SVs^1,W^1)$	2,7069	5,9210	1,3555	1,8962	14,9700	14,9690	14,9700	173 (513)
$(Obj^{0,1,2},SVs^1,W^2)$	2,7057	5,9211	1,3555	1,8963	14,9610	14,9570	14,9390	176 (530)
$(Obj^{0,1,2},SVs^1,W^3)$	2,6132	5,9210	1,3618	1,8962	14,5530	13,9500	12,3730	140 (520)
$(Obj^{0,1,2},SVs^1,W^4)$	2,7056	5,9210	1,3534	1,8962	14,9640	14,9700	14,9690	204 (531)
$(Obj^{0,1,2},SVs^3,W^0)$	2,4418	5,9209	1,5291	1,8957	13,4250	10,3600	6,0680	218 (517)
$(Obj^{0,1,2},SVs^3,W^1)$	2,3933	5,9210	1,6056	1,8954	14,9700	2,6779	8,4052	211 (547)
$(Obj^{0,1,2},SVs^3,W^2)$	2,5164	5,9208	1,3848	1,8951	14,9670	14,0760	7,2919	228 (533)
$(Obj^{0,1,2},SVs^3,W^3)$	2,3424	5,9215	1,6285	1,8960	12,2470	2,6495	8,5043	220 (548)
$(Obj^{0,1,2},SVs^3,W^4)$	2,5598	5,9204	1,3441	1,8941	14,9690	14,9700	9,1597	212 (553)

The 100 against 50 random loops performed in advance needn't tend to be an advantage to achieve better results by the SAM method. The efficiency depends on location of the random design sets in the design space and weight factor specifications. If the random or any design sets, performed before the SAM method is applied, are located mostly in the area of a local extreme the approximation of the dependent variables' functions might considerably influence shape of the objective function in ambient of the global extreme.

The presented optimization problem was also analyzed considering constraints expressed by inequations (9.5.10). The smaller design space in most cases causes lower number of SAM iterations to achieve defined convergence criteria. Summarization of the Obj and DVs' values in the best design sets of the SAM solutions obtained is listed in Tables 9.30 and 9.31 for cases initiated by 50 and 100 random loops separately.

Table 9.30 SAM best design sets in narrower DVs' intervals (init. by 50 random loops)

Settings	**Obj.10^{-1} [m^3]**	**$DV_1.10^{-2}$ [m]**	**$DV_2.10^{-1}$ [m]**	**$DV_3.10^{-1}$ [m]**	**$DV_4.10^{-2}$ [m]**	**$DV_5.10^{-2}$ [m]**	**$DV_6.10^{-2}$ [m]**	**Iterations [-]**
($Obj^{0,1,2}$,SVs^0,W^0)	2,3930	6,0533	1,5843	1,9988	7,2582	9,7072	8,1281	145 (462)
($Obj^{0,1,2}$,SVs^0,W^1)	2,3710	6,0566	1,6529	1,9739	5,2648	9,7877	7,7596	138 (475)
($Obj^{0,1,2}$,SVs^0,W^2)	2,3792	6,0538	1,6374	1,9981	5,4148	10,1350	7,7323	138 (468)
($Obj^{0,1,2}$,SVs^0,W^3)	2,3720	6,0534	1,5894	1,9987	7,5371	8,4457	8,1238	166 (486)
($Obj^{0,1,2}$,SVs^0,W^4)	2,3680	6,0563	1,6571	1,9236	5,3282	10,0370	7,7308	79 (496)
($Obj^{0,1,2}$,SVs^1,W^0)	2,3923	6,0666	1,6506	1,9923	5,3589	10,5210	7,6557	56 (472)
($Obj^{0,1,2}$,SVs^1,W^1)	2,4049	6,0533	1,5905	1,8883	8,2539	10,6500	7,6455	152 (471)
($Obj^{0,1,2}$,SVs^1,W^2)	2,3681	6,0533	1,6271	1,9987	6,0664	8,5095	8,4870	177 (454)
($Obj^{0,1,2}$,SVs^1,W^3)	2,3615	6,0550	1,6240	1,9986	6,4749	6,7406	9,6402	88 (458)
($Obj^{0,1,2}$,SVs^1,W^4)	2,3772	6,0542	1,6500	1,9931	5,0763	10,2220	7,6572	74 (469)
($Obj^{0,1,2}$,SVs^3,W^0)	2,3842	6,1040	1,6611	1,9856	5,0763	10,1740	7,6557	53 (463)
($Obj^{0,1,2}$,SVs^3,W^1)	2,3558	6,0533	1,6647	1,9622	3,8492	8,3961	9,7903	159 (470)
($Obj^{0,1,2}$,SVs^3,W^2)	2,3060	6,0534	1,6681	1,9380	2,1118	10,1230	7,8988	179 (473)
($Obj^{0,1,2}$,SVs^3,W^3)	2,3599	6,0533	1,5665	1,9985	9,0310	6,8480	8,2644	149 (461)
($Obj^{0,1,2}$,SVs^3,W^4)	2,3697	6,0536	1,6494	1,9949	5,1537	9,8000	7,6900	149 (478)

Table 9.31 SAM best design sets in narrower DVs' intervals (init. by 100 random loops)

Settings	Obj.10^{-1} [m^3]	$DV_1.10^{-2}$ [m]	$DV_2.10^{-1}$ [m]	$DV_3.10^{-1}$ [m]	$DV_4.10^{-2}$ [m]	$DV_5.10^{-2}$ [m]	$DV_6.10^{-2}$ [m]	Iterations [-]
($Obj^{0,1,2}$,SVs^0,W^0)	2,3630	6,04710	1,5825	1,9936	8,1993	5,0415	10,6500	111 (546)
($Obj^{0,1,2}$,SVs^0,W^1)	2,3507	6,0624	1,6387	1,7285	8,8848	5,0128	10,8420	154 (517)
($Obj^{0,1,2}$,SVs^0,W^2)	2,3642	6,0559	1,5772	1,9966	8,2020	5,0420	10,8390	114 (531)
($Obj^{0,1,2}$,SVs^0,W^3)	2,3672	6,0547	1,5644	1,9982	8,2011	5,5701	10,7230	231 (560)
($Obj^{0,1,2}$,SVs^0,W^4)	2,3637	6,0581	1,5833	1,9826	8,2042	4,9962	10,8480	152 (524)
($Obj^{0,1,2}$,SVs^1,W^0)	2,3666	6,0534	1,5786	1,9970	8,2033	5,0908	10,8560	115 (506)
($Obj^{0,1,2}$,SVs^1,W^1)	2,3802	6,0533	1,6023	1,7042	10,6990	5,2928	11,0970	216 (511)
($Obj^{0,1,2}$,SVs^1,W^2)	2,3706	6,0533	1,5630	1,9990	8,2042	5,5906	10,8690	206 (506)
($Obj^{0,1,2}$,SVs^1,W^3)	2,3496	6,0534	1,6553	1,7295	7,4508	5,8767	11,0000	176 (523)
($Obj^{0,1,2}$,SVs^1,W^4)	2,3905	6,0533	1,5720	1,7660	11,1700	5,3531	11,0850	222 (513)
($Obj^{0,1,2}$,SVs^3,W^0)	2,3642	6,0539	1,5756	1,9980	8,2017	5,1051	10,8040	132 (510)
($Obj^{0,1,2}$,SVs^3,W^1)	2,3825	6,0534	1,5852	1,7443	10,9440	5,4004	10,8380	259 (535)
($Obj^{0,1,2}$,SVs^3,W^2)	2,3718	6,0534	1,5944	1,7475	10,4860	5,1488	10,8550	135 (521)
($Obj^{0,1,2}$,SVs^3,W^3)	2,3627	6,0538	1,5836	1,9844	8,2028	4,9972	10,7970	126 (515)
($Obj^{0,1,2}$,SVs^3,W^4)	2,3639	6,0534	1,5570	1,9776	9,3830	5,2680	10,0890	258 (551)

Generally, by using a smaller design space (defined by 9.5.10), more equivalent results are obtained and different weight factor definitions don't influence the SAM proceeding as markedly as in the case where greater design space (expressed by 9.5.9) is defined. After all, the best design set, within the frame of the solution by the SAM method performed, is achieved in the solution which is initiated by 100 random loops, defined by greater design space and weight factor is directed to the objective function values.

9.6.5 Discussion and Conclusion

A truss-beam by using optimization methods (First Order and Subproblem Approximation method) was designed. The design was performed according to restrictions of members' resistances stated by EN 1993 Eurocode 3 and minimal allowed first frequency value of the first natural modal shape. This led to two solutions within the frame of each iteration accomplished by the optimization method or tool. The aim of the problem was truss-beam weight minimization subjected to the defined constraints.

At first the First Order Method was applied to design the truss-beam. The design was performed considering different initial design sets, varying gradients' distances in the FOM iterative proceeding and different DVs' limits' ranges according to exploring attitude of the design space performed in advance. The best results were obtained in the cases where the initial design sets are chosen at the bottom of the defined design space. These solutions require more iterations to achieve convergence criteria which requires patience concerning computing time. Furthermore, the analysis of the FOM method demonstrated that a more precise exploration of the design space performed in advance doesn't always achieve more efficient design. A more precise design space exploration increases robustness of the solution at the expense of accuracy. A suitable attitude in searching the objective function extreme by the FOM method is exploring the design space sufficiently to avoid embedding in a local extreme, selecting a suitable step size of gradients and then investigating the best achieved design set ambient which was obtained by the FOM method.

Furthermore, the Subproblem Approximation Method, considering different approximation methods of state variables' functions, approach of weight factor, size of design space and number of initial random loops, was applied within the frame of the truss-beam design. As in the case of design by the FOM method, more precise exploring of the design space before application of the optimization method doesn't necessarily mean achieving better results. A suitable way to find an extreme of the objective function is to sufficiently analyze the design space and localize the estimated global extreme of the problem. If there is a design set in ambient of the global extreme in the database of the optimization process, it is advisable to focus the weight factor on objective function values to improve the solution performed by the SAM method.

Both analyzed methods allow finding an extreme of the objective function with sufficient accuracy. To improve the obtained solutions, it is recommended to analyze best design sets' ambient by the Gradient Tool (section 7.3.4) which allows controlling the sensitivity of the objective function by small changes of each defined design variable (DV).

9.7 EFFICIENT DESIGN OF AIR GAP LOCATION IN WOODEN STUDS

An indispensable factor for structural designs is comfort for living indoors. One of the variables affecting comfort is the temperature. The structure has to be resistant to heat escaping from the building in the winter time, especially in Scandinavian countries, but also resistant to the entry of outdoor heat to the building in summer time. The following case represents an application of optimization algorithms in efficient design of wooden studs, which are widely used in single family houses' timber frames in Finland, to improve their thermal properties by creation a suitable stud's cross-sectional shape. Wood has thermal conductivity of 0,12W/Km. One option to find higher resistivity of studs is to create air gaps inside of them because of the air thermal conductivity which is 0,026W/Km.

The design is limited by conditions stated by manufacturing, stability and construction procedures. With these a suitable way to design an efficient location of the air gaps is using an optimization process. The obtained designs are verified by lab tests and controlled by evaluation of mould growth risk to predict mould existence and possible damage in the future.

9.7.1 Initial Design and Conditions

The initial design of wooden studs' cross section in this analysis is a rectangle with the dimensions of 0,27x0,045m. For achieving higher resistance properties for the studs, air gaps inside of them are created. Restrictions for the problem, stated by manufacturing and construction proceeding and stability requirements, are determined as follows:

- Heat flux should not change in vertical direction. That means the air gaps will be vertical lengthways all over the height of the stud.
- From the manufacturing point of view the air gaps can only be 3 or 5mm wide.
- Manufacturers also require maximal length of air gaps to be 35mm.
- The minimal distance between air gaps can be 10mm.
- The minimal cross-section area of one wooden stud without air gaps can be $0{,}011\text{m}^2$.
- From the construction point of view it is required that the ends of cross-section are 40mm without air gaps.

According to these requirements, the following cross-sections of studs were designed (Figure 9.34). The initial point for the optimization design process is represented by the air gaps' location in the middle of wooden studs' cross-sections.

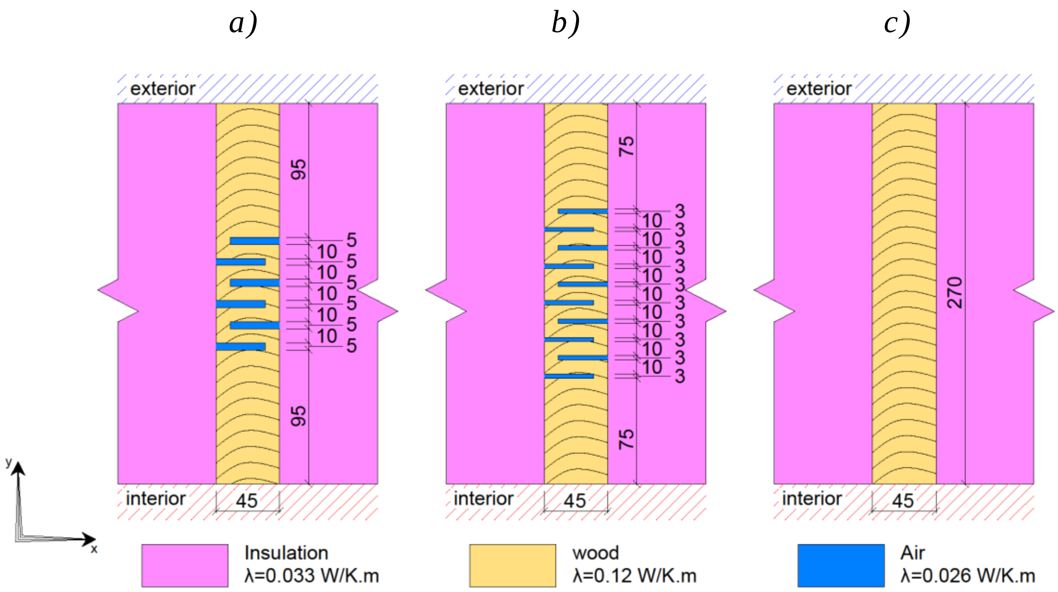

Figure 9.34 Initial design of studs a) six 5mm air gaps, b) ten 3mm air gaps, c) simple wooden stud

9.7.2 Efficient Design of Air Gaps' Location

An optimization process was applied with the aim of finding an efficient location of air gaps inside of a wooden cross-section. The optimization procedure was performed by Subproblem Approximation Method within the frame of the Design Optimization module in the Ansys program. The optimization was represented by minimization of the heat flux value in the middle of outdoor surface of wooden stud, which was changing depending on varying location of air gaps. This, therefore, means that the heat flux value represents an objective function of the optimization process. Independent variables (DVs) are expressed by distances between air gaps and the dependent variable (SV) is represented by the distance of the first air gap from the interior surface of the cross-sections' end. The approximation of objective function was performed by quadratic plus cross-term fitting and state variable by linear curve fitting. The analysis was performed considering varying definition of weighting factors (Table 9.4).

9.7.3 Obtained Results

By applying the optimization technique for the heat flux peak minimization which is achieved in the vicinity of wooden studs on the colder side of the wall, the new air gaps' locations were found. The lowest heat flux peaks were found in the case where air gaps are located as near as possible to the colder side of the wall. In the 3mm air gaps case the heat flux peak is 8,42% lower according to the initial design (Figure 9.35).

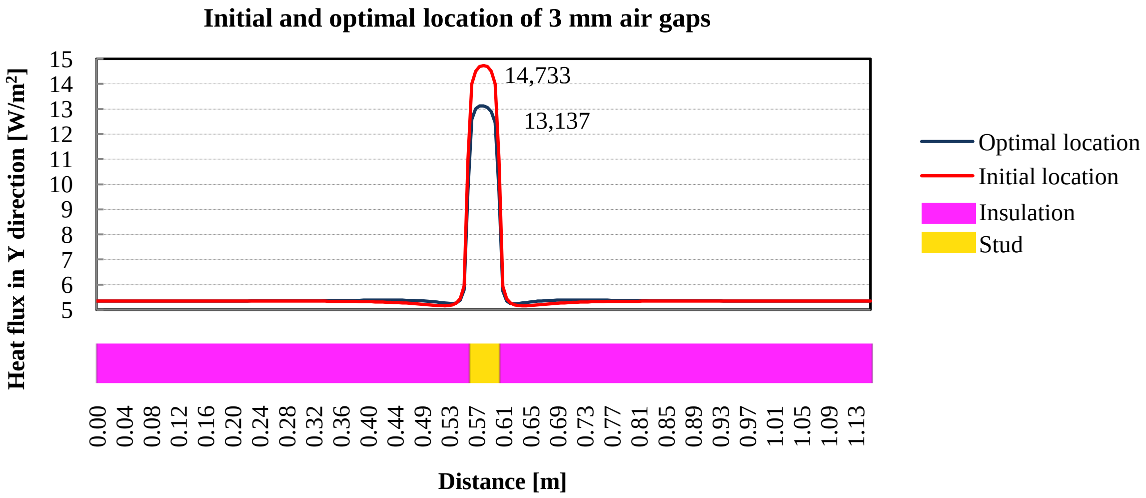

Figure 9.35 Heat flux on colder side of wall in initial and optimized shape of stud with 3mm air gaps

In the case where 5mm air gaps are created inside of the wooden studs the efficient location shows heat flux value as 10,83% lower (Figure 9.36).

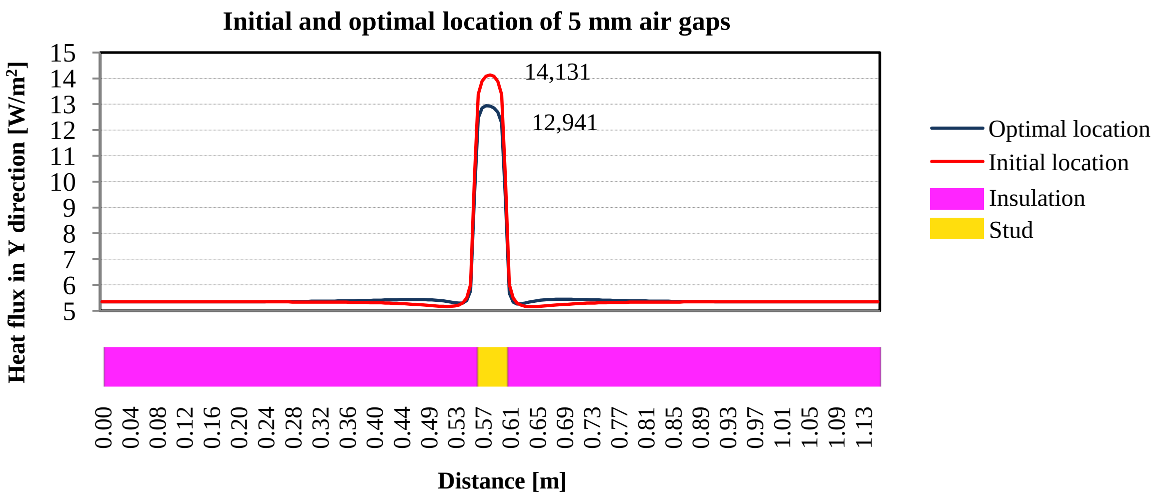

Figure 9.36 Heat flux on colder side of wall in initial and optimized shape of stud with 5mm air gaps

In the case of 3mm air gaps the convergence was achieved in 105th iteration and in the case of 5mm air gaps in 385th iteration where the weights were applied to design sets based on distances in the design space, objective function values and feasibility/infeasibility of obtained design sets in the solution.

It has to be noted that the peak of heat flux values in the wooden stud by varying air gaps' location is changing, but total heat energy outflow keeps the same. This is influenced by behaviour of heat flux in the vicinity of wooden studs in the insulated part of the structure. That means that by lowering heat flux peak in wooden stud the heat flux in the vicinity of the stud increases.

9.7.4 Lab Tests

A total of five wooden studs; three wooden studs with different air gaps' location, the currently used wooden stud and for comparison one simple solid wooden stud were subjugated to measuring of the heat transfer in the laboratory (Figure 9.37). Spaces between studs are filled by insulation. Distance between studs is 555mm, which is the distance obtained from previous calculations, where the heat transfer within the frame of one stud doesn't influence heat transfer of the others. The wooden studs were measured by sensors installed on their surfaces from warm and cold sides. Space on the one side of the lab wall was set at ~20^{0}C and the other ~ -20^{0}C. The model was subjugated to the defined conditions for approximately 4 mounts which is sufficient time to achieve almost steady state simulation of heat transfer.

Figure 9.37 Lab sample - wooden studs with installed measuring sensors

Computed heat transfer data (temperature and heat flux) in the Comsol 4.1a and Ansys were verified with data from measuring sensors obtained in lab tests. The simulations were performed by steady state analyses in two-dimensional models. The differences in verification achieved were in a range of 0,7%, which corresponding to computational model simplifications is considered satisfying. The outcomes from a thermal camera obtained showed bigger differences of temperature values but the trend of thermal characteristics is in the same sense through the lab model. Examples of the thermo camera outcomes are shown in the following picture (Figure 9.38):

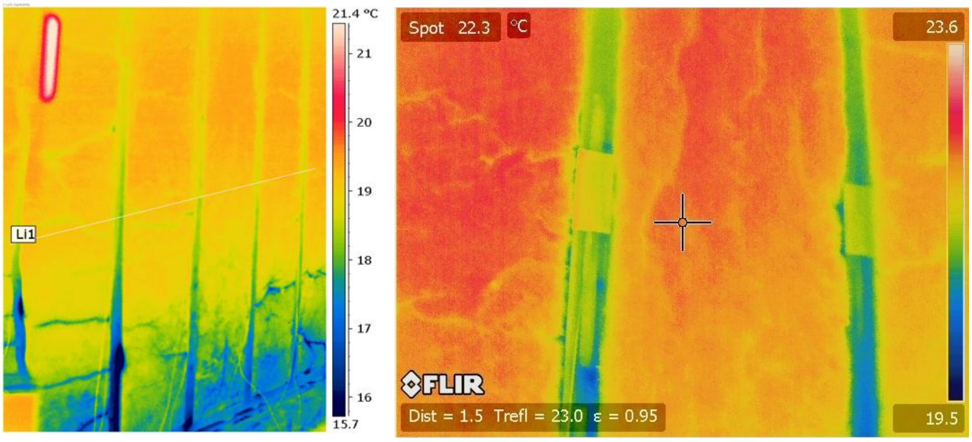

Figure 9.38 Thermal camera outcomes

Based on results in the optimization procedure obtained the most efficient air gaps' location is as near as possible to the outdoor studs' surface (Figure 9.39).

a) b)

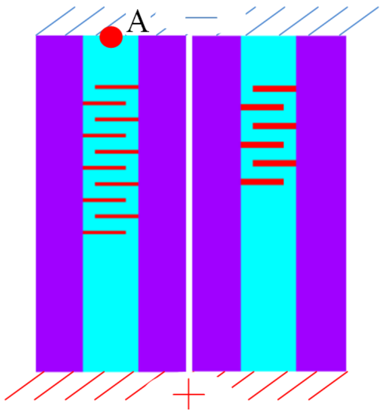

Figure 9.39 Optimized air gaps' location a) ten 3mm air gaps, b) six 5mm air gaps

9.7.5 Discussion and Conclusion

Wooden studs achieve quite high heat flux against a fully insulated wall. The heat flux peak might represent a higher danger of mould growth risk. The efficient design of air gaps' location within a wooden stud is presented. The aim of the work was to minimize the heat flux peak in wooden studs by creating air gaps and maximize their effectiveness by an efficient location. The correlation of computing models was verified with measured data obtained from lab tests.

Two cases were created according to defined restrictions; ten 3mm air gaps and six 5mm air gaps along the wooden studs' cross-section height. The existence of air gaps in the stud achieves better thermal properties compared to solid wooden cross-section. Using an optimization procedure efficient air gaps' locations are defined. Because of heat outflow the work was focused on the outdoor side of the wall. By creating of air gaps as near as possible to the outdoor the peak is lowered by 8,42% in the case with 5mm air gaps and by 10,83% when 3mm air gaps are created against the initial designs, which are represented by the location of air gaps in the middle of wooden studs' cross-sections.

The presented case proved that the air gaps could be a good solution to achieve better results within the frame of heat outflow in stability elements. Further, optimization techniques in Design Optimization module/Ansys were successfully applied to minimize heat flux peak by varying air gaps' location.

The best location of air gaps is near to the colder side of the wall. It means that the optimal location depends on the exterior temperature during the whole year. In case the building is in a location with higher exterior temperature than the interior temperature, the location of air gaps would be the opposite.

The new designs were controlled from the point of view of mould growth risk. According to the theory [50], [68], [69] there is no risk of mould growth in the analyzed structure.

The presented case was achieved within the Pitke and PiRakko projects supported by TEKES and number of Finish prominent companies performed in Oulu University of Applied Sciences.

10 CONCLUSION

Using optimization algorithms in designing new or improving current civil and mechanical engineering problems represents a challenging task for many designers. It requires compactness of mathematical, physical and structural knowledge simultaneously with patience and imagination from the designer.

The aim of the presented study was to investigate user's approachable optimization algorithms widely used in designing new engineering problems or improving existing ones. One of the most used techniques to simulate reality of engineering problems is the Finite Element Method. It allows simulation of structural, chemical and physical phenomena with sufficient faithfulness and accuracy. Improving efficiency of designs performed by systems using the FEM method led developers to implement multi-purpose optimization techniques into their structures.

The specialized field which deals with methodology of optimization techniques is referred to as Operating Research. The widest discipline of Operating Research is mathematical programming which specifies optimization algorithms whose features usually represent foundations for techniques used in efficient or optimal design of engineering problems. Especially within the frame of currently performing researches, great progress arises in modern methods such as genetic algorithms, methods based on neural-networks, simulated annealing, etc. On the other hand the most approachable techniques in practical designing are usually represented by algorithms of linear and nonlinear programming.

Within the frame of the submitted thesis widely used optimization methods applied in technical fields these days are investigated. They are represented by modified mathematical programming techniques to extend their applicability to solving a wide range of optimization problems. The presented methods are First Order and Subproblem Approximation method. Their sequences are based on unconstrained techniques defined by mathematical programming. The methods extract proceedings of unconstrained optimization techniques where problems' constraints are represented by additional terms expressed by penalty functions. Definitions of finite element models which are subjected to optimization procedures are by some means variant. In the course of model creation, parameters which determine optimization variables must be defined. Their definition must correspond to the actual problem by considering changes which will/or might be, performed by the optimization procedure.

The application of the optimization methods in structural designs was performed by means of problems which could be controlled by accurate manually computed and/or graphical solutions, and/or by application of specialized systems such as optiSLang, Matlab, Maple, Mathcad and Comsol. At first, the methods' robustness is tested via a multi-extreme optimization problem. Then its accuracy and ability to find a minimum of one or more variables in strictly convex problems is analyzed. In the following, an efficient design of a truss-beam where the minimum weight is searched, and constraints are defined by EN 1993

Eurocode 3, as well as the minimal allowed first frequency value of the structure's first natural modal shape. The final problem is represented by optimization methods' application in solving a sample of thermal problem by an efficient design of air gap location in wooden studs, which was achieved within the Pitke and PiRakko projects supported by TEKES and number of prominent Finish companies, performed in Oulu University of Applied Sciences. In this case the computed data was verified with outcomes obtained in controlled measuring on laboratory models. The tests achieved confirmed consensus. The analyzed optimization problems achieve satisfactory results, especially in cases where the optimization problem includes few variables, and robustness and accuracy are controllable. The robustness and accuracy of obtained results decreases simultaneously with increase in variables' number.

The available optimization tools allow exploration of design space before an optimization method is applied or improve obtained design sets by investigation of the best solutions' vicinities. The widest coverage of design space for initial exploration could be accomplished by Random Tool which generates random design sets within the frame of determined design space. Definition of problems' constraints is effectively controlled by Factorial Tool which performs design sets in the lower and upper limits of design variables. Sweep Tool generates design sets in directions of design variables initiated by a certain design set. A suitable tool to control the best obtained design set from previous solution is the Gradient Tool which tests sensitivity of dependent variables set by small changes of design variables' values.

First Order Method is generally considered as more accurate at the expense of computing time. The efficiency of a design using the FOM method mostly depends on suitable exploration of a design space, selection of an initial design set and length of gradient defined in each iteration of the optimization procedure. The robustness of the method is not guaranteed, however, there is a possibility to improve it by exploring the design space by one or combination of available optimization tools. As a suitable solution to maximize robustness of the method is even distribution of design sets within the design space performed in advance. If decent exploration of the design space is performed and we assume that the obtained actual best design set is in ambient of the global extreme, short step length definition for solution performed by the FOM method is recommended. By increasing exploring design sets' number the robustness of the solution increases simultaneously.

Subproblem Approximation Method is based on approximations of dependent variables' functions which are then subjected to sequential unconstrained minimization technique to localize an extreme of the objective function. The robustness and accuracy of the method depends on features and number of design sets both performed via exploration of the design space in advance and by own SAM method. In the case that the design sets performed before the SAM method is applied are accumulated further from the global extreme, its location can be easily disregarded. This issue can be slightly controlled by weights applied to the solution performed by the least square technique.

Although both methods are adapted for solution of wide range of technical problems, each optimization problem requires certain approach and investigation. Modelling of the FEM/FEA problem must be designed carefully to avoid contradictions and inaccuracies via the optimization proceeding. An essential part of the optimization methods' application is also suitable determination of optimization variables and specification of optimization problem constraints which are represented by upper and/or lower boundaries of each design and state variable. Before the optimization method is applied, competent design space exploration performed by one of the optimization tool is highly recommended.

The presented optimization methods represent a suitable approach to improve efficient design of a wide range of civil and mechanical engineering structures or elements. By combination of their advantages and FEM/FEA method it is possible to achieve very good results, although robustness of the solutions is not guaranteed. The robustness and accuracy of the procedure could be increased by competent exploration of design space and suitable selections of optimization methods' features.

REFERENCES

[1] ALBOUL, L., VAN DAMME, R.: Optimality criteria and optimization procedure for 2.5D triangulations, University of Twente, Netherlands.

[2] ARKARESVIMUN, R.: Inventory Optimization in a Retail Multi-Echelon Environment, Industrial Engineering Chulalongkorn Universtiy, Massachusetts Institute of Technology, June 2008.

[3] ARORA, J., S.: Introduction to optimum design, Second Edition, University of Iowa, Department of Civil and Environmental Engineering and the Department of Mechanical Engineering, ISBN 0120641550, 2004.

[4] BACON, D., F.: Fast and Effective Optimization of Statically Typed Object-Oriented Language, Universtiy of California, Berkeley, California, Report No. UCB/CSD-98-1017, December 1997.

[5] BAZARAA, M., S., SHERALI H., D., SHETTY, C., M.: Nonlinear programming, Theory and Algorithms, Third Edition, New Jersey, 2006.

[6] BELLMORE, M., GREENBERG, H., J., JARVIS, J., J.: Generalized Penalty-function Concepts in Mathematical Optimization, Operations Research - pgs. 229-252, USA, 1970.

[7] BERGE, J., M., F.: Least Squares Optimization in Multivariate Analysis, University of Groningen, Leiden, Netherlands, 2005.

[8] BLÁHA, M.: Maximalizace tuhosti rovinného tělesa při daném zatížení a daném objemu, DP, České vysoké učení technické v Praze, Fakulta strojní, Ústav mechaniky, biomechaniky a mechatroniky, Odbor pružnosti a pevnosti, Prague 2006.

[9] BOYD, S., KIM, S., J., VANDENBERGHE, L., HASSIBI, A.: A tutorial on geometric programming, DOI 10.1007/s11081-007-9001-7, USA, 2007.

[10] BOYD, S., VANDENBERGHE, L.: Convex Optimization, *publisher by the press syndicate of the University of Cambridge*, Cambridge, 2004, ISBN 0 521 83378 7.

[11] CHINNECK, J., W.: Practical Optimization: A Gentle Introduction, Carleton University, Systems and Computer Engineering, Canada, Ottawa 2007.

[12] CHONG, E., K., P., ZAK, S., H.: An Introduction to Optimization, Third Edition, Wiley-Interscience Series in Discrete Mathematics and Optimization, New York, ISBN 9780471758006, 2008.

[13] CHUNG, W., SESSIONS, J. (department of Forest Engineering, Oregon State University, United States of America), HEINIMANN, R. H. (Swiss Federal Institute of Technology, Zurich, Switzerland): Optimization of cable harvesting equipment placement and road locations using digital terrain models.

[14] COTTLE, R., LEMKE, C., E.: Nonlinear Programming: Nonlin Program Siams 9P, American Mathematical Society, New York, ISBN: 0821813293, 1976.

[15] CURTIS, R. VOGEL: Computational Methods for Inverse Problems, Montana State University Bozeman, Montana, 2002.

[16] DAHL, G.: Introduction to optimization, CMA, Dept. of Mathematics and Dept. of Informatics, University of Oslo, 2008.

[17] DANTZIG, G., B.: Linear Programming, Department of Management Science an Engineering, Stanford University, Stanford, California, 2002.

[18] DANTZIG, G., B., THAPA, M., N.: Linear Programming, 2: Theory and Extensions, Springer Series in Operations Research, ISBN: 0-387-98613-8, 2006.

[19] DONG, S.: Methods for Constrained Optimization, 18.086 Project 2, Spring, 2006.

[20] FERGUSON, T., S.: Linear Programming, Mathematics Department, University of California at Los Angeles.

[21] FERGUSON, T., S.: Optimal Stopping and Applications, Mathematics Department, University of California at Los Angeles.

[22] FILOMENO COELHO, R.: Optimization based on evolutionary algorithms for aeronautics, Cenaero, with support of the Walloon Region and European Structural Funds ERSF, ESF.

[23] FRANDSEN, P., E., JONASSON, K., NIELSEN, H., B., TINGLEFF, O.: Unconstrained Optimization, 3. Edition, Technical University of Denmark, Informatics and Mathematical Modelling, March, 2004.

[24] FRODL, M.: Genetické algoritmy a jejich aplikace, Úvod do umělé inteligence, 2009.

[25] GASS, S., I.: Linear Programming, Methods and Applications, ISBN: 048643284X, 2003.

[26] GEDIG, M., STIEMER, S.: Decision Tools for the Engineering of Steel Structures, electronic Journal of Structural Engineering, 6, 2006.

[27] GOVAN, A.: Introduction to Optimization, SAMSI NDHS Undergraduate workshop, North Carolina State University, 2006.

[28] HAVRDA, J.: Matematické programování, SNTL – Nakladatelství technické literatury, Prague, 1972.

[29] HERNÁNDEZ, S., FONTÁN, A. N., PEREZZÁN, J. C., LOSCOS, P.: Design optimization of steel portal frames, School of Civil Engineering, University of Coruna, Spain, 2005.

[30] HOUŽVIČKA, R,: Optimalizace topologie ohýbaných desek, DP, České vysoké učení technické v Praze, Fakulta strojní, Ústav mechaniky, biomechaniky a mechatroniky, Odbor pružnosti a pevnosti, Prague 2006.

[31] HUEBNER, K., H., DEWHIRST, D., L., SMITH, D., E., BYROM, T., G.: The Finite Element Method for Engineers, Fourth Edition, New York, Chichester, Weinheim, ISBN 0 471 37078 9, 2001.

[32] IVANENKO, A. S.: Optimality principle for nondegenerate grids, Computing Center of the Russian Academy of Sciences, Moscow, Russia.

[33] JAMRÓZ, T.: Návrh topologie prutové a rámové konstrukce, DP, České vysoké učení technické v Praze, Fakulta strojní, Ústav mechaniky, biomechaniky a mechatroniky, Odbor pružnosti a pevnosti, Prague 2004.

[34] KALL, P., WALLACE, S., W.: Stochastic Programming, Second Edition, 1994.

[35] KLAPKA, J., DVOŘÁK, J., POPELA, P.: Metody operačního výzkumu, VUTIUM Brno, 1996, ISBN: 80-214-1839-7.

[36] KLAPKA, J., DVOŘÁK, J.: Úvod do operační a systémové analýzy, Vysoké učení technické v Brně, Fakulta strojního inženýrství, Brno, 2002.

[37] KOCINA, K.: Využití optimalizačních algoritmů při návrhu konstrukcí, bakalářská práce, Fakulta stavební, Ústav stavební mechaniky, Brno 2009.

[38] KRUMKE, O., S.: Integer Programming - Polyhedra and Algorithms, Technishe Universität Kaiserslautern, Germany, 2006.

[39] KUMAR, N.: Optimization Methods: Advanced Topics in Optimization – Direct and Indirect Search, Bangalore, Kanada.

[40] LEVY, A., B.: The Basics Of Practical Optimization, January 12, 2009.

[41] LIŠKA, M.: Metody operačního výzkumu, distanční opora, Ostravská univerzita, Přírodovědecká fakulta, Ostrava, 2002.

[42] LUKŠAN, L.: Numerické optimalizační metody, učební text – revidovaná verze, Institute of Computer Science, Academy of Sciences of the Czech Republic, 2005.

[43] MAHFOUZ, S. Y.: Design optimization of structural steelwork - abstract, University of Bradford, United Kingdom, 1999.

[44] MAIXNER, L.: Matematické programování, Přírodovědecká fakulta Univerzity Palackého, Olomouc, 1980.

[45] MAREŠ, T.: Maximalizace tuhosti laminátových desek a trubek volbou orientace vrstev, disertační práce, Fakulta strojní, Ústav mechaniky, České vysoké učení technické v Praze, Prague, 2004.

[46] MAREŠ, T.: Základy konstrukční optimalizace, Tomáš Mareš, Prague, 2006, ISBN 80 239 6508 5.

[47] MILLER, S., J.: The Method of Least Squares, Mathematics Department, Brown University, USA.

[48] MOLER, C.: Numerical Computing with Matlab, Society for Industrial and Applied Mathematics, MathWorks, USA, 2008.

[49] NOCEDAL, J., WRIGHT, S. J.: Numerical Optimization, Springer-Verlag New York, 1999, ISBN 0 387 98793 2.

[50] OJANEN, T., PEUHKURI, R., VIITANEN, H., LÄHDESMÄKI, K., VINHA, J., SALMINEN, K.: Classification of material sensitivity - New approach for mould growth modelling, 9th Nordic Symposium on Building Physics - NSB 2011, ISBN 978 952 15 2573 5, Tampere - FI, 2011.

[51] OSYCZKA, A., MRZYGLOD, M.: Evolutionary Optimization of Mechanical Structures in Computer Simulated Environment, 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, 30 May – 03 June, Brazil, 2005.

[52] PEDREGAL, P.: Introduction to Optimization, texts in applied mathematics, Springer, ISBN 0387403981, 2004.

[53] POLI, R., LANGDON, B., W., MCPHEE, F., N., KOZA R., J.: A field guide to genetic programming, ISBN 978 1 4092 0073 4, UK - USA, March 2008.

[54] PRESS, W., H., Teukolsky, S., A., Flannery, B., P., Vetterling, W., T.: Numerical Recipes: The art of scientific computing, Cambridge University, ISBN: 0521880688, 2007.

[55] RAO, S. S.: Engineering Optimization – Theory and Practise, Third Edition, School of Mechanical Engineering, Purdue University, John Wiley & Sons, Indiana, 1996.

[56] REN, Q.: Optimizing Behavioral Transformations using Taylor Expansion Diagrams, Electrical and Computer Engineering, University of Massachusetts Amherst, February, 2008.

[57] RIEDEL, R.: Efektivní implementace simplexového algoritmu do knihovny OOSol, bakalářská práce, Fakulta elektrotechniky a informatiky, katedra aplikované matematiky, Ostrava, 2008.

[58] RŮŽIČKOVÁ, I., FAJMON, B.: Matematika 3, Fakulta elektrotechiky a komuikačích technologií VUT v Brně, Brno, 2003.

[59] SAARIO, A.: Mathematical Modeling and Multiobjective Optimization in Development of Low-Emission Industrial Boilers, Tampere University of Technology, Tampere, 2008.

[60] SCHIJNDEL, V., J.: Integrated Modeling using MatLab, Simulink and COMSOL, VDM Verlag Dr. Muller Aktiengesellschaft & Co. KG, U.S.A., 2008, ISBN: 978-3-639-10669-5.

[61] SHOPOVA, G., E., VAKLIEVA-BANCHEVA, G., N.: BASIC - A genetic algorithm for engineering problems solution, Institute of Chemical Engineering, Bulgarian Academy of Sciences, March 2006.

[62] SILVEIRA, R., A., M., PEREIRA, W., L., A., CONCALVES, P., B.: Constrained and Unconstrained Optimization Formulations for Structural Elements in Unilateral Contact with an Elastic Foundation, Research Article, Hindawi Publishing Corporation, Mathematical Problems in Engineering, Brazil 2008.

[63] SMITH, A., E., Coit, D., W.: Penalty Functions, Handbook of Evolutionary Computation, Department of Industrial Engineering, University of Pittsburgh, USA, 1995.

[64] STEPHENS, A., C., HANNA, G., J., GONG, L.: Thermal-Fluid Analysis of the Fill and Drain Operations of a Cryogenic Fuel Tank, National Aeronautics and Space Administration, USA, December 1993.

[65] ŠTĚPÁNEK, P.: Optimalizační úlohy v inženýrských úlohách: Výzvy a požadavky, Pravděpodobnost porušování konstrukcí – PPK 2006.

[66] TAUFER, I., HRUBINA, K., TAUFER, J.: Algoritmy a algoritmizace, Vývojové diagramy, Pardubice-Prešov-Hradec Králové, January 2001.

[67] VANDERBEI, R., J.: Linear Programming, Foundations and Extension, Second Edition, ISBN: 0-7923-7342-1, 2001.

[68] VIITANEN, H., OJANEN, T.: Improved Model to Predict Mold Growth in Building Materials, VTT Technical Research Centre of Finland, 8 pages., Finland, 2007.

[69] VIITANEN, H., TORATTI, T., MAKKONEN, L., PEUHKURI, R., OJANEN, T., RUOKALAINEN, L., RÄISÄNEN, J.: Towards modelling of decay risk of wooden materials, VTT Technical Research Centre of Finland, DOI 10.1007/s00107-010-0450-x, Finland, 2010.

[70] VÍTEČKOVÁ, M., JEDLIČKA, D.: Statická optimalizace systémů, Vysoká škola báňská – Technická univerzita Ostrava, Fakulta strojní, Katedra automatizační techniky a řízení, 2003.

[71] VÝVODA, L.: Timošenkova metoda při řešení kmitání laminátových trubek, DP, České vysoké učení technické v Praze, Fakulta strojní, Ústav mechaniky, biomechaniky a mechatroniky, Odbor pružnosti a pevnosti, Prague 2006.

[72] WANG, X.: Method of Steepest Descent and its Applications, Department of Engineering, University of Tennessee, Knoxville, USA, 2008.

[73] WEISE, T.: Global Optimization Algorithms – Theory and Application, e-book, 2009.

[74] YUAN, Y.: A New Stepsize for the Steepest Descent Method, Academy of Mathematics and System Sciences, Chinese Academy of Science, Beijing, China, 2004.

STANDARDS AND USER MANUALS

[75] ANSYS Release 11.1 Documentation Preview.

[76] ANSYS Design Optimization Seminar, for Revision 5.0, Swanson Analysis, System, inc.

[77] COMSOL, Optimization lab, user’s guide, version 1.2., 2008.

[78] EN 1993 - Eurocode 3: Design of steel structures.

ABBREVIATIONS

g, h, w	*Inequality constraints*
l	*Equality constraints*
i	*Number of design condition, i.e* i^{th} *design condition*
m	*Total number of inequality constraints*
p	*Total number of equality constraints*
f	*Objective function (Obj)*
n	*Number of variables*
k	*Number of monomial*
K	*Total number of monomials*
U	*Objective function of dynamic programming*
u	*Partial objective function of dynamic programming*
t	*Transformation function of dynamic programming*
v	*Input and output data (state variables) of dynamic programming*
Z	*Set of integers*
∇	*Gradient*
α, β, γ	*Tolerances of design conditions*
j	*Number of iteration*
k	*Total number of iterations*
e	*Vertical distance of point from function*
E	*Squared error*
s	*Step size*
P	*Penalty function*
q	*Penalty parameter*
ε, ρ, τ	*Convergence and penalty criteria (tolerance)*
ϕ	*Weight of function*
F	*Unconstrained objective function*
X	*Penalty function of design variable*
W	*Penalty function of state variable*
x	*Optimization variables*
Δ	*Difference*
N	*Maximum number of iterations*

Vectors and Matrices

$\boldsymbol{x}$	*Vector of design variables*
$\boldsymbol{H}$	*Hessian matrix*
$\boldsymbol{K}$	*Stiffness matrix*
$\boldsymbol{C}$	*Damping matrix*
$\boldsymbol{u}$	*Displacement vector*
$\boldsymbol{f}$	*Force vector*
$\boldsymbol{M}$	*Mass matrix*
$\ddot{\Delta}$	*Nodal acceleration vector*
$\dot{\Delta}$	*Nodal velocity vector*
Δ	*Nodal displacement vector*

Subscripts

w	*Number of monomial*
f	*Feasible solution*
u	*Infeasible solution*
r	*Number of random iterations*
s	*Number of sweep evaluations*
fa	*Number of factorial evaluations*
g	*Number of gradient evaluations*

Superscripts

a	*Exponent of monomial*
γ	*Multiplier of monomials exponent*
$*$	*„Optimum“, the best from obtained solutions*
λ	*Exponent of penalty function*

Accents

$\overline{\ }$	*Upper limit of design variable*
$\underline{\ }$	*Lower limit of design variable*
$\widehat{\ }$	*Approximation*

LIST OF FIGURES

LIST OF TABLES

CURICULUM VITAE

PERSONAL DATA

FILIP FEDORIK

Tuulihaukantie 6 B 28
902 50 Oulu
Finland

E-mail: fedorikfilip@gmail.com
Tel: +358 (0)449 107 107

Born: 11.01.1982, Děčín, Czech Republic
Nationality: Czech

EDUCATION

09/2006 - present	Doctoral study programme, Institute of Structural Mechanics, Brno University of Technology, Brno, Czech Republic Ph.D. Thesis: "Using Optimization's Algorithms by Designing of Structures".
09/2001 - 06/2006	Master of Science, Brno University of Technology, Brno, Czech Republic Thesis: "Bridge Deck Cross Section Case Study with Regard to Effect of Wind" (in Czech).

WORK EXPERIENCE

09/2009 - 12/2012	Project Designer, The Oulu Region Joint Authority for Education, Oulu University of Applied Sciences, School of Engineering, Oulu, Finland
02/2009 - 07/2009	Technology Trainee, The Oulu Region Joint Authority for Education, Oulu University of Applied Sciences, School of Engineering, Oulu, Finland

PARTICIPATION IN RESEARCH PROJECTS

01/2012 - present	GACR 104/11/0703 Use of Progressive Materials in Cyclically Stressed Structures (Použití progresivních materiálů u cyklicky namáhaných konstrukcí).
08/2010 - 12/2012	IEEB: Increasing Energy Efficiency in Buildings, Funded by EU Interreg 4A Nord programme.
06/2011 - 12/2012	PiRakko: Pientalojen Rakentamisen Aikainen Kosteusmittaustutkimus (Research and Measuring of Moisture in Small Houses), Funded by Kansallinen TEKES and about 10 companies.
02/2009 - 07/2011	Pitke: House Building Design and Production Quality Modelling and Testing, Funded by TEKES.
01/2007 - 12/2008	Solved Numerical Examples Into the Subject Dynamics of Structures, Project supported of the Ministry of Education, Youth and sports in Czech Republic.

TEACHING EXPERIENCE

09/2009 - 12/2012	Guest lecturer at School of Engineering, Oulu University of Applied Sciences. Subjects: Fundamentals of FEM/FEA method, Fundamentals of using the COMSOL program, Introduction to Heat and Moisture analysis
09/2006 - 06/2008	Lectured at the Institute of Structural Mechanics, Faculty of Civil Engineering, Brno University of Technology, Czech Republic. Subjects: Fundamentals of Structural Mechanics, Structural Analysis I, Structural Analysis II (in Czech)

INTERNATIONAL EXPERIENCES

10/2008 - 08/2009	Oulu University of Applied Sciences, School of Engineering, Erasmus (exchange programme), Oulu, Finland

LANGUAGES

Czech (native fluency), English, Finnish (basic)

AWARDS

06/2006	Award of Dean (for Master's thesis), Brno University of Technology (BUT).
05/2006	3rd place in international students scientifically competition, Brno University of Technology, Faculty of Civil Engineering, Institute of Structural Mechanics. Theme: "Interference Forces from Wind on Bridge Deck by the Change Geometry" (in Czech).

AUTHORS PUBLICATIONS

[1] FEDORIK, F.: Design Optimization Algorithms Applied in Structural Designing, Wulfenia Journal, IF: 0,216, Vol.20, No.8, Klagenfurt, Austria, 2013, ISSN: 1561-882X.

[2] FEDORIK, F.: Analysis of Gradient and Approximation Design Optimization Methods' Robustness, 19th International Conference on Soft Computing - Mendel 2013, June 26-28, 2013, Brno, Czech Republic, ISBN: 978-80-214-4755-4, ISSN: 1803-3814.

[3] FEDORIK, F., ILLIKAINEN, K.: Wooden vs. Concrete Blocks' Structure - HAM and Mould Growth Analysis, SB13 Oulu - Sustainable Procurement in Urban Regeneration and Renovation, Oulu, Finland, May 22-24, 2013, ISBN: 978-951-758-562-0, ISSN: 0356-9403.

[4] FEDORIK, F., ILLIKAINEN, K.: HAM and Mould Growth Analysis of a Wooden Wall, 5th Nordic Passive House Conference: Passivhus Norden 2012, Trondheim, Norway, 2012, ISBN: 978-823-210-173-3.

[5] FEDORIK, F.: Efficient Design of Air Gap Location in Wooden Studs, 10th Inernational Conference on New Trends in Statics and Dynamics of Buildings, Bratislava, Slovakia, 2012, ISBN: 978-80-227-3786-9.

[6] FEDORIK, F.: Varied-Colour Roof Surface & Their Influence on a Structure, 10th Inernational Conference on New Trends in Statics and Dynamics of Buildings, Bratislava, Slovakia, 2012, ISBN: 978-80-227-3786-9.

[7] FEDORIK, F.: Možnosti využití optimializačních modulů v systému Ansys, Konstrukce (journal), vol. 4, 2008, ISSN: 1213-8762.

[8] FEDORIK, F.: Výpočet účinnosti aerodynamických pohlcovačů na válci, International Conference 70 Years of FCE STU, Bratislava, Slovakia, 2008, ISBN 978-80-227-2979-6.

[9] FEDORIK, F.: Využití optimalizačních metod při řešení stability ocelových konstrukcí, International Conference 70 Years of FCE STU, Bratislava, Slovakia, 2008, ISBN 978-80-227-2979-6.

[10] KALA, J., SALAJKA, V., FEDORIK, F.: Wind induced harmonic forces on steel cylinder with various strake systems, The 4th International Conference on Advances in Wind and Structures (AWAS'08), Jeju, Korea, 2008, ISBN 978-89-89693-23-9.

[11] FEDORIK, F.: Možnosti využití optimalizačních modulů v systému Ansys, Juniorstav - 10th Professional Conference of Postgraduate Students, Faculty of Civil Engineering, Brno, Czech Republic, 2008, ISBN: 978-80-86433-45-5.

[12] FEDORIK, F.: Volba vhodného tvaru příčného profilu mostovky vzhledem k statickým účinkům větru, Juniorstav - 9th Technical Conference for Doctoral Study Faculty of Civil Engineering, Brno, Czech Republic, 2007, ISBN: 80-214-2828-7.

SUBMITTED

[13] FEDORIK, F., ILLIKAINEN, K.: HAM and Mould Growth Analysis of a Wooden Wall, International Journal of Sustainable Built Environmental (IJSBE), 2013, ISSN: 2212-6090.

[14] FEDORIK, F.: Efficient Design of a Truss Beam by Applying First Order Optimization Method, ICNAAM 2013 - 11th International Conference of Numerical Analysis and Applied Mathematics, Rhodes, Greece, September 21-27, 2013.

[15] FEDORIK, F.: Gradient vs. Approximation Design Optimization Techniques in Low-Dimensional Convex Problems, ICNAAM 2013 - 11th International Conference of Numerical Analysis and Applied Mathematics, Rhodes, Greece, September 21-27, 2013.